



TUGAS AKHIR - TE 141599

**RANCANG BANGUN PROTOTIP
WIRELESS SENSOR UNTUK MENGATUR
TINGKAT KETINGGIAN AIR SUNGAI**

Burhanudin Rabani
NRP 2211100079

Dosen Pembimbing
Dr. Muhammad Rivai, ST., MT.
Suwito, ST., MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT - TE 141599

**DESIGN OF PROTOTYPE WIRELESS SENSOR
FOR MANAGING RIVER HEIGHT'S LEVEL**

Burhanudin Rabani
NRP 2211100079

Advisor
Dr. Muhammad Rivai, ST., MT.
Suwito, ST., MT.

DEPARTEMENT OF ELECTRICAL
ENGINEERING
Faculty Of Industrial Technology
Sepuluh Nopember Institute Of Technology

**RANCANG BANGUN PROTOTIP *WIRELESS SENSOR* UNTUK
MENGATUR TINGKAT KETINGGIAN AIR SUNGAI**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

Pada

**Bidang Studi Elektronika
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing I,

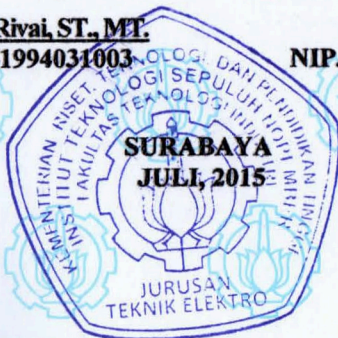
Dosen Pembimbing II,

Dr. Muhammad Rivai, ST., MT.

NIP. 196904261994031003

Suwito, ST., MT.

NIP. 198101052005011004



RANCANG BANGUN PROTOTIP WIRELESS SENSOR UNTUK MENGATUR TINGKAT KETINGGIAN AIR SUNGAI

Nama : Burhanudin Rabani
NRP : 2211100079
Jurusan : Teknik Elektro
Dosen Pembimbing : 1. Dr. Muhammad Rivai. ST., MT.
2. Suwito, ST., MT.

ABSTRAK

Sungai merupakan jalan air mulai dari hulu menuju ke hilir. Air sungai ini bisa bermanfaat dan bisa menimbulkan bencana. Bencana yang dapat dihasilkan adalah banjir dikarenakan ketinggian air sungai yang melebihi batas badan sungai. Saat ini sungai mempunyai banyak pintu air yang dapat mengatur ketinggian air sungai untuk mencegah bencana dan menambah manfaat sungai bagi manusia.

Wireless Sensor merupakan suatu kumpulan sensor yang dapat berkomunikasi tanpa menggunakan kabel (wireless), mengukur besaran fisis, memproses data dan dapat ditempatkan di area tertentu. Wireless sensor terdiri dari sensor ultrasonik untuk mengukur ketinggian air sungai, transmitter dan receiver untuk komunikasi data serta mikrokontroler untuk memproses data. Tujuan penelitian ini adalah dapat mengatur tingkat ketinggian air sungai dengan menggunakan komunikasi yang nirkabel sebagai koordinasi antara pintu-pintu air. Sehingga dapat menghindari banjir yang disebabkan oleh sungai dan dapat menambah manfaat dari sungai.

Hasil percobaan menunjukkan wireless sensor dapat saling berkomunikasi sejauh 40 meter. Kecepatan pintu air terbuka dan tertutup pada kecepatan maksimal sebesar 2 mm per detik. Pengaturan ketinggian air sungai dalam simulasi dengan menggunakan metode On/Off mempunyai error rata-rata keseluruhan pada saat pengujian pada prototip sebesar 5.35 mm.

Harapan dalam penelitian ini dapat diterapkan di pintu air untuk mengatur ketinggian air secara otomatis dan mencatat data ketinggian air sungai secara akurat.

Kata Kunci: *Ketinggian Air Sungai, Sensor Ultrasonik, Wireless Sensor.*

DESIGN OF PROTOTYPE WIRELESS SENSOR FOR MANAGING RIVER HEIGHT'S LEVEL

Name : Burhanudin Rabani
NRP : 2211100079
Department : Electrical Engineering
Adivsor : 1. Dr. Muhammad Rivai, ST., MT.
2. Suwito, ST., MT.

ABSTRACT

The river is the water from upstream toward downstream. The river water can be useful and could lead to disaster. Flooding is disaster that can be generated because the river water level exceed the maximum river height level. Currently the river has many floodgates that can regulate the water level of the river to prevent disasters and to add river's benefits for human.

Wireless Sensor is a set of sensors that can be communicate wirelessly (wireless), measure physical quantities, processing data and can be placed in a certain area. Wireless sensors consist of ultrasonic sensors to measure the water level of the river, the transmitter and receiver for data communication and a microcontroller to process the data. The purpose of this study is to set the level of the river water levels by using wireless communication as coordination between the doors of the water. So as to avoid flooding caused by rivers and can add to the benefits of the river.

The results showed wireless sensors can communicate with each other as far as 40 meters. The speed of floodgates opening and closing at a maximum speed is 2 mm in a second. Setting the water level of the river in the simulation by using the On / Off method has the overall average error of testing on the prototypes at 5.35 mm.

In the future, this study can be applied in the floodgates to regulate water levels automatically and record data accurately river water level.

Keywords:: *Ultrasonik Sensor, Water Height Level, Wireless Sensor.*

KATA PENGANTAR

Alhamdulillah, syukur yang tiada henti penulis panjatkan kehadiran Allah SWT., serta tidak lupa sholawat salam tetap tercurah kepada junjungan kita Nabi Muhammad SAW. sehingga penelitian dalam tugas akhir ini bisa berjalan lancar. Selama pelaksanaan penelitian Tugas Akhir ini, penulis mendapatkan bantuan dari berbagai pihak, dan penulis ingin menyampaikan rasa terima kasih. Diantara yang membantu dalam selesainya tugas akhir ini adalah :

1. Keluarga yang senantiasa memberikan do'a, nasihat, motivasi, dan dukungan.
2. Bapak Dr. Muhammad Rivai, ST., MT. sebagai pembimbing pertama, atas bimbingan, pengarahan, ide dan motivasi yang diberikan selama pengerjaan tugas akhir ini.
3. Bapak Suwito, ST., MT. selaku dosen pembimbing kedua atas bimbingan, ide, pengarahan, dan motivasi yang diberikan selama pengerjaan penelitian tugas akhir ini.
4. Bapak Ir. Tasripan, MT. selaku dosen penulis atas nasehat dan bimbingannya selama penulis kuliah di Jurusan Teknik Elektro ITS.
5. Bapak dan Ibu dosen penguji penulis yang telah memberikan kritik, saran dan masukan yang membangun.
6. Bapak dan Ibu dosen ITS, khususnya Jurusan Elektro, terutama Bidang Studi Elektronika yang telah membimbing dan mengajar penulis.
7. Bapak Mahendra Sitinjak A.Md beserta rekan rekan dari Perum Jasa Tirta 1 yang telah membantu berupa data dan pengalaman selama mengerjakan tugas akhir ini.

Penulis sadar bahwa Tugas Akhir ini masih belum sempurna dan masih banyak hal yang perlu diperbaiki. Saran, kritik dan masukan baik dari semua pihak sangat membantu penulis terutama untuk berbagai kemungkinan pengembangan lebih lanjut.

Surabaya, Juni 2015

Burhanudin Rabani

DAFTAR ISI

ABSTRAK.....	i
ABSTRACT.....	iii
KATA PENGANTAR.....	v
Daftar Isi.....	vii
Daftar Gambar.....	xi
Daftar Tabel.....	xiii
BAB I PENDAHULUAN	
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	2
1.3. Batasan Permasalahan.....	2
1.4. Tujuan.....	3
1.5. Metodologi.....	3
1.6. Sistematika Penulisan.....	4
1.7. Relevansi.....	5
BAB II DASAR TEORI	
2.1. Wireless Sensor.....	7
2.2. Sensor Ultrasonik HC-SR04.....	9
2.2.1 Prinsip Kerja Sensor.....	10
2.3. Arduino Uno.....	12
2.4. Serial Komunikasi.....	14
2.5. Pulse Width Modulation.....	15
2.6. Sungai.....	17
2.7. Pintu Air.....	17
2.8. Driver Motor L298.....	18
2.9. RF Transmitter dan Receiver 433 Mhz.....	22
BAB III PERANCANGAN SISTEM	

3.1. Perancangan <i>Base Station</i>	26
3.1.1 Perancangan Perangkat Keras <i>Base Station</i>	26
3.1.2 Diagram Alir Kerja <i>Base Station</i>	28
3.1.3 Perancangan Perangkat Lunak <i>Base Station</i>	30
3.1.3.1. Program <i>Virtual Wire</i>	30
3.1.3.2. Program Pengiriman Data Ke Komputer.....	31
3.1.3.3. Program Kontroler On/Off.....	33
3.1.3.4. Program Delphi.....	34
3.2. Perancangan Node Wireless Sensor	35
3.2.1 Perancangan Perangkat Keras <i>Node Wireless Sensor</i>	35
3.2.2 Diagram Alir Wireless Sensor	37
3.2.3 Perancangan Perangkat Lunak <i>Wireless Sensor Node</i>	40
3.2.3.1. Pembacaan Sensor Ultrasonik.....	40
3.2.3.2. Implementasi Aktuator di Driver Motor L298.....	41
3.3. Perancangan Suplai Daya	43
3.4. Perancangan Prototip Pintu Air Sungai	44
BAB IV PENGUJIAN SISTEM	
4.1. Pengujian Sensor Ultrasonik HC-SR04	49
4.2. Pengujian Jarak Komunikasi Antar <i>Node</i> dan <i>Base Station</i> ..	52
4.3. Pengujian Suplai Daya	54
4.4. Pengujian Motor DC	55
4.5. Pengujian Kontrol On/Off	58
BAB V PENUTUP	
5.1. Kesimpulan.....	63
5.2. Saran.....	63
Daftar Pustaka.....	65
Lampiran.....	67

program base station.....	67
program node A wireless sensor.....	75
program node B wireless sensor.....	80
Riwayat Hidup Penulis.....	89

DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino UNO.....	13
Tabel 3.1 Klasifikasi Ketinggian Air.....	33
Tabel 3.2 Klasifikasi Ketinggian Air.....	34
Tabel 4.1 Pengujian Sensor HC-SR04.....	41
Tabel 4.1 Pengujian Sensor HC-SR04 (lanjutan).....	41
Tabel 4.2 Pengujian Jarak dan Error.....	53
Tabel 4.3 Karakteristik Motor DC pada Pintu 1.....	55
Tabel 4.4 Karakteristik Motor DC pada Pintu 2.....	57

DAFTAR GAMBAR

Gambar 1.1	Pintu Air Gunung Sari Surabaya.....	1
Gambar 2.1	Blok diagram Wireless Sensor.....	8
Gambar 2.2	Sensor Ultrasonik HC-SR04.....	10
Gambar 2.3	Pantulan Gelombang Ultrasonik pada Objek.....	11
Gambar 2.4	Diagram Waktu Sensor Ultrasonik.....	12
Gambar 2.5	Arduino Uno.....	13
Gambar 2.6	Perangkat Lunak Arduino.....	14
Gambar 2.7	Duty Cycle pada PWM.....	16
Gambar 2.8	Gambar Skematik Modul L298.....	20
Gambar 2.9	Pin Masukan dan Keluaran pada Modul L298.....	21
Gambar 2.10	L298 Modul Driver Motor.....	22
Gambar 2.11	Modul Pengirim (kiri) dan Penerima (kanan) RF 433Mhz.....	23
Gambar 3.1	Diagram Blok Sistem.....	25
Gambar 3.2	Diagram Base Station.....	27
Gambar 3.3	Perangkat Keras Base Station.....	28
Gambar 3.4	Diagram Alir Base Station.....	30
Gambar 3.5	Grafik Penampil Data pada Delphi.....	34
Gambar 3.6	Diagram Node Wireless Sensor.....	36
Gambar 3.7	Peletakan Aktuator dan Sensor.....	37
Gambar 3.8	Diagram Alir Node Wireless Sensor.....	39
Gambar 3.9	Skema Rangkaian Suplai Daya.....	43
Gambar 3.10	Suplai Daya.....	44
Gambar 3.11	Desain Pintu Air.....	46
Gambar 3.12	Desain Pintu Air Keseluruhan.....	47
Gambar 4.1	Grafik Pengukuran Sensor Ultrasonik terhadap Ketinggian Air.....	51
Gambar 4.2	Gambar Pengukuran Sensor Ultrasonik terhadap Ketinggian Air.....	52
Gambar 4.3	Grafik Error Pada Komunikasi.....	53
Gambar 4.4	Pengukuran Tegangan Suplai Daya	54
Gambar 4.5	Grafik Kecepatan Naik dan Turun Pintu Air 1.....	56
Gambar 4.6	Grafik Kecepatan Naik dan Turun Pintu Air 2.....	58
Gambar 4.7	Grafik Ketinggian pada Pintu Air 1.....	59
Gambar 4.8	Grafik Ketinggian pada Pintu Air 2.....	60
Gambar 4.9	Grafik Ketinggian pada Pintu Air 1 pada Pengujian Keseluruhan.....	61

Gambar 4.10	Grafik Ketinggian pada Pintu Air 2 pada Pengujian Keseluruhan.....	62
-------------	--	----

BAB I

PENDAHULUAN

1.1. Latar Belakang

Sungai adalah aliran air yang besar dan memanjang yang mengalir secara terus menerus dari hulu (sumber) menuju hilir (muara) dan biasanya dibuat oleh alam. Sungai sangat penting untuk kehidupan manusia. Sungai mempunyai banyak manfaat seperti pengairan sawah, minum, sumber air dan lain lain. Aliran air sungai dipengaruhi oleh curah hujan disekitar sungai, aktivitas manusia, sumber air di hulu sungai dan tingkat pasang air laut di hilir sungai. Namun sungai juga dapat memberikan bencana bagi manusia seperti adanya banjir, pencemaran, dan penyebaran penyakit. Peristiwa yang paling banyak terjadi adalah banjir, dimana banjir ini ditandai dengan tingginya ketinggian air sungai sehingga sungai tidak mampu menampung dan akhirnya meluap. Banjir adalah peristiwa terbenamnya daratan karena volume air yang meningkat. Peningkatan volume air yang membanjiri daratan banyak dipengaruhi oleh curah hujan dan melelehnya es. Volume air yang meningkat ini dialirkan dari dataran tinggi menuju laut melalui sungai-sungai yang sudah tercipta oleh alam maupun yang dibuat sendiri oleh manusia. Air yang terdapat di sungai-sungai mempunyai banyak manfaat untuk memenuhi kebutuhan manusia sebagai contoh untuk mengaliri lahan persawahan, air minum, mandi, mencuci dan lain-lain.



Gambar 1.1 Pintu Air Gunung Sari Surabaya

Ketinggian air sungai ini dapat diatur dengan adanya pintu-pintu air yang dapat mengatur ketinggian air di sungai tersebut. Pengaturan ketinggian air sungai dengan cara mengatur aliran air yang melewati pintu air. Dalam sepanjang aliran sungai dalam satu wilayah terdapat satu hingga beberapa pintu air.

Pintu air saat ini masih menggunakan pintu air yang di fungsikan secara manual mulai dari pengaturan bukaan pintu air, pencatatan ketinggian pintu air hingga komunikasi antar pintu air masih menggunakan tenaga manusia dan berkomunikasi dengan radio *Handy Talkie* sedangkan pengatur pintu air masih menggunakan saklar. Hal ini sangat rentan akan terjadinya kelalaian manusia dalam pengaturan pintu air sehingga perlu di buat suatu sistem untuk dapat membantu kinerja manusia agar dapat menghindari terjadinya kelalaian.

Wireless Sensor adalah suatu jaringan komunikasi nirkabel yang dapat berkomunikasi satu sama lain untuk saling tukar menukar data serta mengolah data dari besaran yang disensor oleh sensornya . Selain itu, *Wireless Sensor* juga dapat ditambahkan aktuator untuk implementasi hasil pengolahan data untuk mengatur ketinggian air.

Oleh karena itu itu diajukanlah tugas akhir ini dengan harapan dapat mengatur ketinggian air sungai secara otomatis dengan menggunakan *Wireless Sensor* .

1.2. Perumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan beberapa masalah, antara lain :

1. Bagaimana cara *node wireless sensor* dapat mengukur ketinggian air sungai
2. Bagaimana cara *node wireless sensor* dapat saling berkomunikasi
3. Bagaimana cara mengumpulkan data dari *node wireless sensor*
4. Bagaimana cara mengatur ketinggian air sungai

1.3. Batasan Permasalahan

Untuk menyelesaikan masalah dalam Tugas Akhir ini, maka perlu diberi batasan - batasan dan asumsi sebagai berikut:

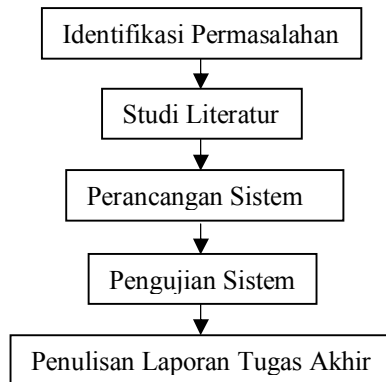
1. *Node wireless sensor* yang digunakan berjumlah 2 buah
2. Menggunakan 1 pintu air dalam setiap bendungan

3. Menggunakan frekuensi 433 Mhz

1.4. Tujuan

Tujuan yang ingin dicapai dalam perancangan ini adalah merancang suatu *node wireless sensor* yang dapat mengirim dan menerima data, mengukur ketinggian air sungai, membuat server untuk mengumpulkan data, mengatur pintu air.

1.5. Metodologi



IDENTIFIKASI PERMASALAHAN

Pada tahapan ini penulis mengidentifikasi masalah yang bersangkutan dengan *node wireless sensor* pengaturan ketinggian air sungai pada pintu air.

STUDI LITERATUR

Hal-hal yang dipelajari dalam tahap studi literatur, meliputi :

1. Mempelajari *node wireless sensor*.
2. Mempelajari pengaturan ketinggian air sungai pada pintu air.

PERANCANGAN SISTEM

Sistem yang dirancang merupakan sebuah perangkat keras dan perangkat lunak *node wireless sensor* yang terdiri dari *node* dan *base station*. *Node wireless sensor* digunakan untuk mengambil data

ketinggian air sungai dan untuk mengatur actuator. *Base station* digunakan untuk mengumpulkan data dari *Node wireless sensor*, memproses data, memberikan perintah pada *node wireless sensor* dan menampilkan data yang diperoleh di komputer.

PENGUJIAN SISTEM

Ada beberapa pengujian yang dilakukan dalam tugas akhir ini, yaitu :

1. Pengujian jarak komunikasi antara *base station* dan *node wireless sensor*
2. Pengujian pengukuran ketinggian air
3. Pengujian *pulse width modulation*
4. Pengujian kontrol On/Off
5. Pengujian suplai daya

PENULISAN LAPORAN TUGAS AKHIR

Tahap ini adalah tahapan terakhir dari proses pengerjaan tugas akhir ini. Tahap ini dimulai saat pengambilan data. Laporan tugas akhir ini berisi tentang semua kegiatan yang dilakukan selama mengerjakan tugas akhir.

1.6. Sistematika Penulisan

Dalam penulisan buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika penulisan sebagai berikut:

BAB I : PENDAHULUAN

Pada bagian ini menjelaskan tentang dasar penyusunan tugas akhir yang terdiri dari beberapa sub bagian, yaitu latar belakang, permasalahan dan batasannya, tujuan yang diharapkan, metodologi pengerjaan tugas akhir, relevansi serta sistematika penulisan tugas akhir ini.

BAB II : DASAR TEORI

Bagian ini menjelaskan tentang dasar teori yang digunakan dalam pengerjaan tugas akhir ini.

BAB III : PERANCANGAN SISTEM

Pada bagian ini berisi perancangan sistem yang membahas mengenai perancangan perangkat keras dan perangkat lunak *node wireless sensor* dan *base station*

BAB IV : PENGUJIAN

Pada bagian ini akan menjelaskan mengenai cara kerja sistem dan analisisnya.

BAB V : PENUTUP

Bagian ini merupakan bagian akhir yang berisikan kesimpulan yang diperoleh dari pembuatan tugas akhir ini, serta saran-saran untuk pengembangannya.

1.7. Relevansi

Matakuliah yang mendukung tugas akhir ini adalah komunikasi data, dasar sistem telekomunikasi serta sensor dan actuator.

Hasil dari tugas akhir ini diharapkan dapat diimplementasikan pada Pintu Air .

BAB II

DASAR TEORI

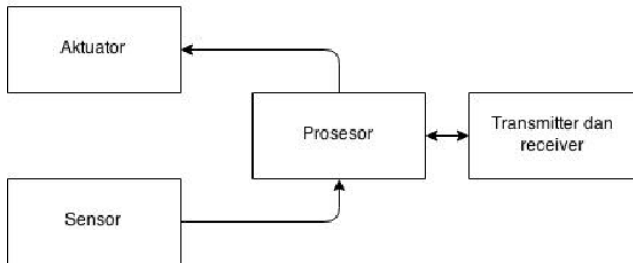
Dasar teori ini berisi tentang pembahasan mengenai sungai, pintu air dan komponen-komponen penyusun *node wireless sensor* seperti arduino uno, modul pemancar dan penerima *radio frequency* 433 Mhz serta sensor jarak ultrasonik HC-SR04. Sedangkan metode kontrol yang digunakan metode kontrol on/off.

2.1. Wireless Sensor

Sensor adalah suatu alat yang dapat merespon suatu stimulus dan memproduksi sinyal listrik yang sesuai dengan nilai yang diterapkan oleh stimulus[1]. Masalah lain dalam diskusi tentang sensor di perbedaan antara sensor pasif dan aktif sensor. Aktif sensor adalah sensor yang memerlukan sumber tegangan ac maupun dc dari luar. Sebagai contoh sebuah sensor tekanan memerlukan 7.5 V dc untuk dapat aktif. Namun tanpa tegangan eksternal tersebut maka sensor tidak dapat mengeluarkan keluaran apapun. Sedangkan sensor pasif adalah sensor yang mempunyai tenaga yang diperoleh dari fenomena yang diperoleh.

Sensor mempunyai sensitifitas yang didefinisikan sebagai sebuah perbedaan karakteristik sensor dengan kondisi ideal. Sensor yang lain sensitifitas didefinisikan sebagai keluaran tegangan yang diberikan pada input. Sensor juga mempunyai jarak nilai yang bisa diukur minimum dan maksimum.

Wireless sensor adalah sensor yang mempunyai pengiriman data pengukuran dari sensor tanpa menggunakan media kabel. Perangkat yang digunakan dalam pengiriman data melalui *wireless* dapat menggunakan sinyal radio, wi-fi, internet dan lain lain.



Gambar 2.1 Blok diagram *wireless sensor*

Sensor dalam *wireless sensor* digunakan untuk mengukur besaran yang ada dalam jangkauan sensor tersebut. Data dari sensor dapat berupa data digital, komunikasi serial, tegangan analog, frekuensi. Sensor-sensor yang dapat dipasang dalam suatu *wireless sensor* yang dapat dipakai adalah ultrasonik, suhu, kelembapan dan lain lain. Sensor-sensor ini mengubah besaran fisik menjadi besaran listrik yang dapat dibaca oleh prosesor.

Prosesor dalam *wireless sensor* adalah suatu rangkaian yang dapat memproses sinyal dan data serta mengatur seluruh kegiatan yang ada dalam wireless seperti mengambil data dari sensor, melakukan komunikasi dan mengatur aktuator. Pemrosesan sinyal yang terjadi dalam prosesor adalah dengan sinyal dari sensor yang dapat berupa analog, digital, komunikasi serial, frekuensi dan lain-lain dapat diubah dalam bentuk tertentu yang dapat diolah lebih lanjut misalnya jika sinyal keluaran sensor berupa tegangan analog maka tegangan analog ini dapat diubah ke bentuk digital dengan menggunakan konversi *analog to digital converter*. Hasil konversi yang berupa digital akan dapat diproses lebih lanjut oleh prosesor yang ada. Pemrosesan data dari *wireless sensor* dapat menggunakan beragam cara tergantung dari harga, performa dan konsumsi daya yang digunakan.

Terdapat berbagai jenis prosesor yang digunakan untuk mengontrol kinerja dari *wireless sensor*, salah satunya adalah mikrokontroler. Mikrokontroler banyak digunakan karena memiliki konsumsi daya yang rendah dan efisien dalam menjalankan program. Sebagai perbandingan bila menggunakan *digital signal processor* (DSP) yang mempunyai harga mahal dan berukuran besar. Namun, DSP bekerja baik apabila digunakan untuk memproses sinyal dengan algoritma yang spesifik.

Prosesor selanjutnya adalah dengan *Field Programmable Gate Array* (FPGA). FPGA lebih cepat dibandingkan dengan mikrokontroller dan DSP serta FPGA mendukung komputasi paralel. FPGA dalam *sensor* yang dapat mengukur besaran melalui sensor, memproses dan berkomunikasi dalam satu waktu dapat dijalankan dengan baik. Namun FPGA harga dan kesulitan dalam memprogramnya membuat FPGA kurang diminati [2].

Selanjutnya dalam *wireless sensor* terdapat modul komunikasi yang berfungsi untuk melakukan pengiriman atau penerimaan data. Data dikirimkan dalam gelombang elektromagnetik dalam udara.

2.2. Sensor Ultrasonik HC-SR04

Sensor ultrasonik adalah sensor jarak yang memanfaatkan pantulan gelombang ultrasonik dengan frekuensi 40Khz. Selain digunakan untuk mengukur jarak sensor ultrasonik ini digunakan untuk mendeteksi objek yang ada didepanya [3]. Sensor ini terdiri dari rangkaian pemancar ultrasonic pada rangkaian pemancar dan penerima. Pemancar memancarkan gelombang ultrasonic sedangkan penerima menerima hasil pantulan gelombang ultrasonic yang dipancarkan oleh pemancar..

Gelombang ultrasonik adalah gelombang mekanik yang memiliki frekuensi diatas 20 khz. Gelombang ultrasonic dapat merambat melalui zat padat, cair maupun gas. Gelombang ultrasonik merambat melalui udara dengan kecepatan 344 meter per detik, mengenai objek dan memantul kembali ke sensor ultrasonic. Gelombang ultrasonic hanya bisa di dengar oleh makhluk tertentu seperti kelelawar dan ikan paus. Kelelawar menggunakan gelombang ultrasonik untuk berburu di malam hari dan ikan paus menggunakannya untuk berenang di dalam lautan yang gelap.

Sensor ultrasonik HC-SR04 bekerja pada tegangan 5 Volt dan mengkonsumsi arus sebesar 15 mA . Jarak yang dapat diukur 0.02 – 4 meter dengan ketelitian 3 mm.



Gambar 2.2 Sensor Ultrasonik HC-SR04

2.2.1 Prinsip Kerja Sensor

Sensor ultrasonik mempunyai bagian transmitter gelombang ultrasonik dan receiver gelombang ultrasonik. Gambar 2.3 transmitter berfungsi untuk mengeluarkan gelombang ultrasonik yang nantinya gelombang ultrasonik ini akan terpantul pada sebuah objek. Objek ini akan memantulkan gelombang ultrasonik agar dapat kembali ke receiver ultrasonik. Receiver ultrasonik akan menerima gelombang ultrasonik yang telah dipantulkan dan mengkonversikannya menjadi sinyal listrik.

Sensor ultrasonik HC-SR04 mempunyai 4 pin input dan output yaitu pin catu daya 5V, *trigger pulse input*, *echo pulse input*, *ground*. Pin catu daya 5V dan *ground* digunakan untuk memberikan daya ke sensor ultrasonik agar dapat bekerja. *Trigger pulse input* digunakan untuk memberikan tanda pada ultrasonik untuk melakukan pengukuran. *Echo pulse input* digunakan sebagai tanda sinyal ultrasonik yang dipancarkan terpantul kembali atau tidak.

Sensor ultrasonik HC-SR04 dapat mengukur jarak objek dengan cara memberikan input pada *trigger pulse input* selama 10 μ s selanjutnya sensor akan mengeluarkan 8 pulsa yang mempunyai frekuensi 40 kHz yang dikeluarkan pada transmitter ultrasonik yang dapat terlihat pada gambar 2.4. Setelah selesai pada sinyal pulsa ke 8 lalu pin *echo pulse* berubah dari tegangan *low* (tegangan 0 Volt) menjadi *high* (tegangan 5 Volt). Keadaan pin *echo pulse* akan berubah kembali ke *low* apabila sinyal 40kHz yang dipancarkan telah kembali ke sensor atau pada waktu tertentu apabila sinyal tidak kembali ke sensor. Waktu *pin echo pulse* ketika pada tegangan *high* yang akan diukur untuk dikonversikan menjadi jarak dengan rumus.

$$S = \frac{t * v}{2} \quad (2.1)$$

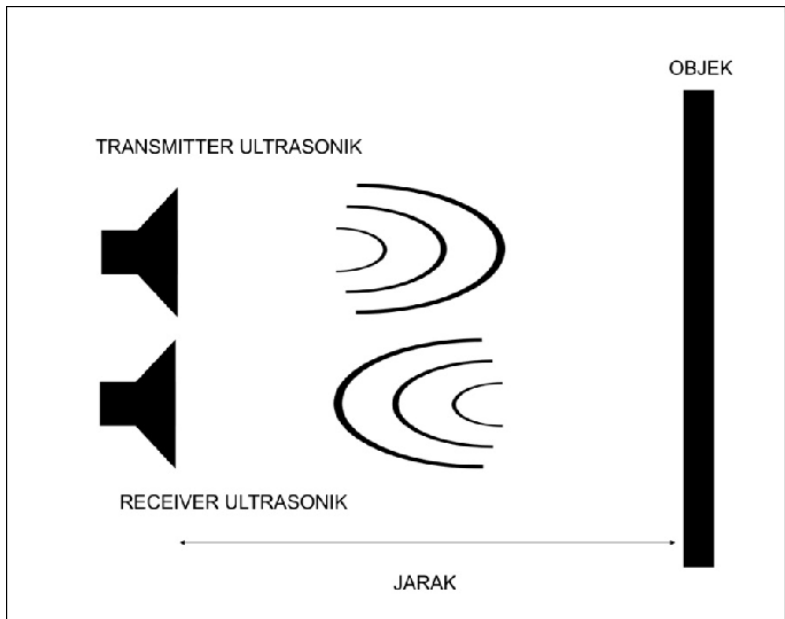
Dimana :

S = jarak yang terukur (meter)

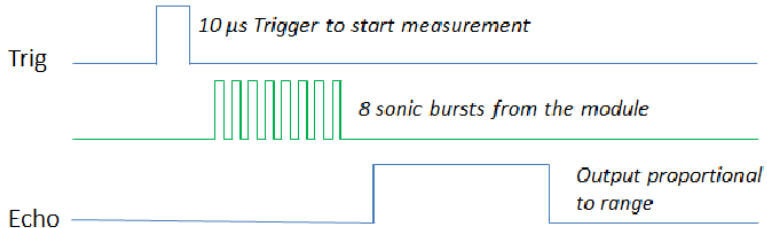
t = waktu *echo pin pulse high* (mikro detik)

v = cepat rambat suara dalam udara (340 meter per detik)

Sehingga jarak dari sensor menuju objek dapat terukur dengan memperhatikan waktu *high pin echo input*. Selanjutnya waktu tersebut dapat di proses untuk mendapatkan jarak yang telah diukur.



Gambar 2.3. Pantulan Gelombang Ultrasonik pada Objek



Gambar 2.4. Diagram Waktu Sensor Ultrasonik

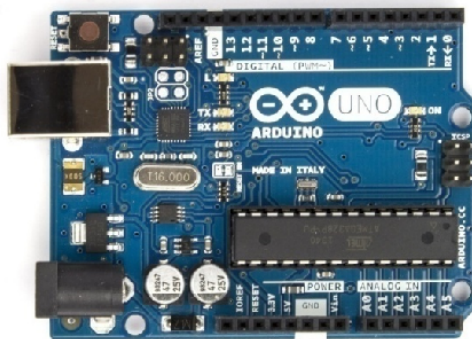
2.3. Arduino Uno

Arduino Uno adalah salah satu produk berlabel arduino yang sebenarnya adalah suatu papan elektronik yang mengandung mikrokontroler ATmega328 (sebuah keping yang secara fungsional bertindak seperti sebuah computer). Peranti ini dapat dimanfaatkan untuk mewujudkan rangkaian elektronik dari yang sederhana hingga yang kompleks. Pengendalian LED hingga pengontrolan robot dapat diimplementasikan dengan menggunakan papan berukuran relatif kecil[4].

Uno berarti satu dalam bahasa Italia dan diberi nama untuk menandai peluncuran Arduino 1.0. Arduino UNO adalah board berbasis mikrokontroler ATmega328. Arduino UNO sendiri mempunyai 14 pin digital *input/output* (6 pin digunakan sebagai keluaran PWM), 6 masukan analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah *ICSP header*, dan sebuah tombol *reset*. Arduino UNO sendiri memuat segala hal yang menunjang mikrokontroler, mudah terhubung dengan komputer dengan sebuah kabel USB ataupun dengan suplai adaptor AC ke DC atau menggunakan baterai. Berikut merupakan spesifikasi arduino UNO

Tabel 2.1 Spesifikasi Arduino UNO

Mikrokontroler	Atmega328
Tegangan pengoperasian	5V
Tegangan input yang disarankan	7-12V
Batas tegangan input	6-20V
Jumlah pin I/O digital	14 (6 menyediakan keluaran PWM)
Jumlah pin masukan analog	6
Arus DC tiap pin I/O	40 mA
Arus DC untuk pin 3.3V	50 mA
Memory flash	32 KB (ATMega328) 0.5 Kb digunakan oleh <i>bootloader</i>
SRAM	2 KB (ATMega328)
EEPROM	1 KB (ATMega328)
Kecepatan	16 MHz
Panjang	68.6 mm
Lebar	58.4 mm
Berat	25 gram



Gambar 2.5 Arduino Uno

Arduino UNO dapat deprogram dengan *software* Arduino. Arduino memiliki sifat *Open Source* yang memudahkan untuk menulis kode dan meng-upload ke *board* Arduino. Perangkat lunak ini dapat berjalan pada

Windows, Mac OS X, dan Linux. Berikut merupakan tampilan dari perangkat lunak arduino



Gambar 2.6 Perangkat Lunak Arduino

Kelebihan menggunakan arduino pemrogramanya yang relatif mudah, perangkat lunaknya yang bersifat terbuka sehingga banyak *library* yang bisa dipakai dan perangkat keras yang open source sehingga dapat dimodifikasi sedemikian rupa hingga sesuai dengan kebutuhan pengguna.

2.4. Serial Komunikasi

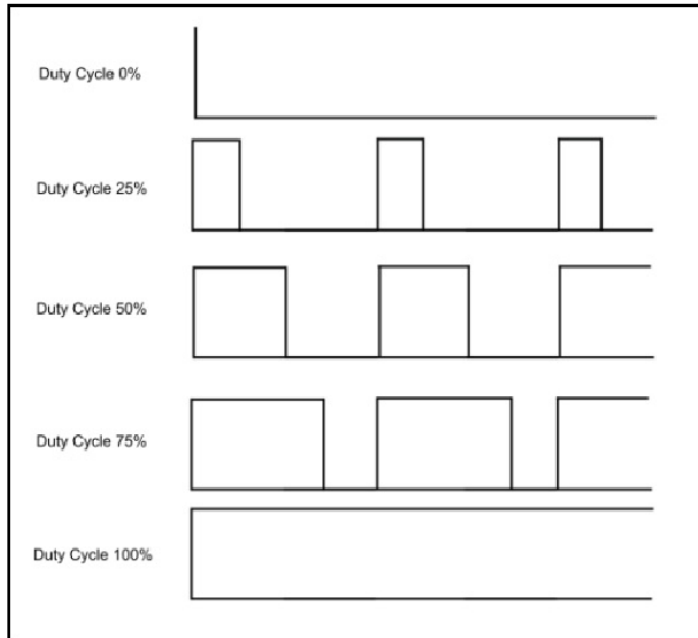
Arduino uno mempunyai komunikasi serial sebagai komunikasi sebagai komunikasi antara mikrokontroler dengan perangkat yang lain misalnya komputer ataupun perangkat mikrokontroler lainnya. Komunikasi serial memiliki 2 pin yaitu pin TX dan pin RX yang terdapat

pada pin 0 dan 1 pada arduino. Pin TX digunakan untuk mengirim data dan pin RX digunakan untuk menerima data.

Pengiriman data serial mempunyai dua macam yaitu sinkron dan asinkron. Pengiriman data sinkron adalah pengiriman data yang terjadi sinkronisasi pewaktuan diawal pada bagian penerima dan pengirim. Pengiriman secara asinkron adalah pengiriman data secara langsung tanpa ada sinkronisasi peawaktuan antara pengirim dan penerima namun dalam pengiriman data terdapat isyarat untuk bit awalan blok data dan bit akhiran blok data serta adanya *baud rate* yang menentukan berapa kali bit terkirim perdetik contohnya *baud rate* 9600 yang menandakan terdapat 9600 bit yang terkirim perdetik. Bit awalan blok data diawali dengan tegangan 0 volt atau logika 0 sedangkan pada bit akhiran blok data diakhiri dengan tegangan 5 volt atau logika 1. Pada bagian blok data terdiri dari 5 – 8 bit data.

2.5. Pulse Width Modulation

Pulse width modulation (PWM) adalah teknik manipulasi lebar pulsa suatu sinyal digital pada frekuensi tertentu. PWM digunakan untuk mendapat suatu tegangan analog dari suatu perangkat digital. Tegangan analog adalah tegangan 0 hingga 5 volt pada suatu perangkat digital yang mempunyai nilai 0 dan 5 volt sehingga tegangan yang keluar pada perangkat digital tidak hanya 0 dan 5 volt namun bisa tegangan antara 0 hingga 5 volt sebagai contoh 2 volt, 3,5 volt dan lain sebagainya. Tegangan analog ini di dapat dengan cara memperhitungkan waktu lebar pulsa 5 volt dan waktu lebar pulsa 0 volt. Nilai perbandingan antara waktu lebar pulsa 5 volt dan waktu lebar pulsa 0 volt dalam 1 periode disebut *duty cycle*.



Gambar 2.7 Duty Cycle pada PWM

Pulse width modulation dalam arduino uno terdapat pada pin 3,5,6,9,10 dan 11 dimana pada pin 3,9,10 dan 11 mempunyai frekuensi sekitar 490 Hertz dan pada pin 5 dan 6 mempunyai frekuensi hingga 980 Hertz [4]. PWM pada arduino di tentukan oleh nilai 0 – 255 yang menentukan *duty cycle* dimana 0 berarti nilai selalu 0 Volt dan 255 berarti 5 Volt sehingga setiap nilai perubahan 1 pada nilai arduino maka dengan menggunakan persamaan

$$V_{pwm} = \frac{\text{nilai_PWM}}{255} \times 5 \quad (2.2)$$

Dimana

V_{pwm} : nilai tegangan keluaran dari pin PWM arduino

Nilai_PWM : nilai PWM pada arduino antara 0-255

Mempunyai perubahan tegangan 0.0196 Volt. Sehingga dapat di membuat sinyal analog dengan ketelitian hingga 0.0196 Volt.

2.6. Sungai

Sungai adalah sistem yang terus menerus dari hulu ke hilir dan muara. [5]. Sungai mempunyai manfaat untuk irigasi, bahan baku air minum, saluran pembuangan air hujan, bahkan berpotensi untuk wisata air sungai.

Sungai mempunyai banyak jenis menurut debit airnya yaitu sungai permanen yaitu sungai yang debitnya sepanjang tahun relative tetap contohnya sungai Kapuas, sungai Barito dan sungai Musi. Selanjutnya sungai periodik yaitu sungai yang ketika musim hujan airnya banyak dan ketika musim kemarau airnya sedikit contohnya sungai Brantas, sungai Bengawan Solo. Jenis sungai yang terakhir adalah sungai episodik yaitu sungai yang mengalirkan air ketika musim hujan dan tidak mengalirkan air saat musim kemarau.

Ketika debit air meningkat atau berkurang menyebabkan ketinggian air sungai menjadi naik atau turun. Ketinggian air sungai ini harus selalu dijaga pada ketinggian tertentu agar tidak terjadi sesuatu hal yang merugikan manusia. Apabila ketinggian air sungai menjadi tinggi melebihi batas yang ditentukan maka akibatnya akan menyebabkan banjir yang sangat merugikan manusia. Banjir ini dapat menyebabkan penyakit, merusak infrastruktur misalnya jalan, gedung, taman dan lain-lain. Namun jika ketinggian air ini kurang dari yang ditentukan maka volume air akan sedikit sehingga pasokan air untuk masyarakat menjadi berkurang.

2.7. Pintu Air

Mengatur ketinggian air agar menjadi sesuai yang ditentukan maka di buatlah pintu air. Pintu air digunakan untuk mengatur aliran air dengan cara membuka atau menutup pintu air. Sehingga ketika pintu air terbuka maka aliran air akan mengalir dengan deras dan dapat menurunkan ketinggian air pada sisi hulu sungai dan meningkatkan ketinggian air pada sisi hilir sungai. Sebaliknya ketika pintu air tertutup maka aliran air akan sedikit sehingga menyebabkan ketinggian air pada sisi hulu akan meningkat dan menurunkan ketinggian air pada sisi hilir.

Naik turunya pintu air dapat digerakkan dengan berbagai cara yaitu dengan cara manual maupun otomatis. Cara manual adalah dengan cara memutar penggerak pintu air dengan menggunakan tenaga manusia

secara langsung. Sedangkan cara otomatis bisa dengan menggunakan motor sebagai pengeraknya.

2.8. Driver Motor L298

L298 adalah dua buah penyearah penuh yang di desain untuk menerima standard tegangan logika TTL dan dapat mengendalikan beban induktif seperti relay, motor DC dan motor Stepper. L298 mempunyai 2 input yang diinginkan untuk mengaktifkan atau menonaktifkan alat secara independen dari sinyal input. Emitter dari transistor terbawah dari setiap penyearah tersambung bersama dan berhubungan dengan terminal luar dan dapat digunakan sebagai tahanan perasa luar.

Driver motor L298 terdiri dari input tegangan mulai dari 7 volt hingga 46 volt. Input tegangan ini dapat langsung di koneksikan ke sebuah sumber tegangan arus searah. Driver motor L298 terdapat beberapa pin yang digunakan untuk mengontrol kecepatan maupun arah dari putaran motor. Pin yang digunakan untuk mengontrol kecepatan dan putaran arah putaran motor yaitu bernama pin EnA, In1, In2 untuk mengontrol motor yang pertama dan EnB, In3, In4 untuk mengontrol motor yang kedua.

EnA dan EnB adalah masukan dari driver motor L298 yang digunakan untuk mengatur kecepatan motor DC. Pin EnA dan EnB terdapat pada huruf I dan D pada gambar 2.8 dan gambar 2.9. Pengaturan kecepatan ini menggunakan tegangan yang dimasukkan pada pin EnA dan pin EnB. Tegangan masukan ini dapat diproduksi dari PWM pada mikrokontroler, pembagi tegangan dan *adjustable power supply*. Keluaran PWM dalam mikrokontroler khususnya arduino berupa tegangan mulai dari 0 Volt hingga 5 Volt. Pin EnA digunakan untuk mengontrol kecepatan motor pada motor 1 sedangkan EnB digunakan untuk mengontrol kecepatan motor pada motor 2.

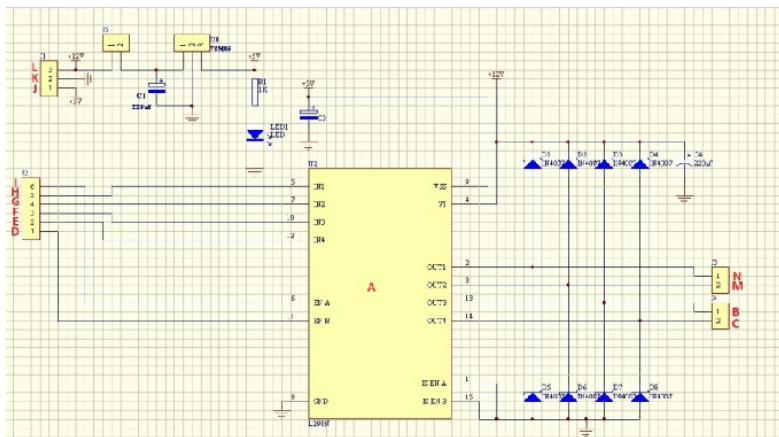
Arah dari putaran motor ditentukan oleh pin In1, In2, In3 dan In4. Pin In1 dan In2 dapat mengontrol arah putaran dari motor 1 sedangkan pin In3 dan In4 dapat mengontrol arah putaran dari motor 2. Rangkaian yang ditunjukkan pada gambar 2.10 input L298 berupa In1, In2, In3 dan In4 merupakan input digital pada IC L298 yang menjadi masukan dari gerbang AND sehingga pin In1, In2, In3 dan In4 serta pin EnA dan EnB akan mengaktifkan transistor H-Bridge.

Transistor H-Bridge adalah susunan transistor yang susunannya membentuk huruf H. Rangkaian pada gambar 2.10 rangkaian H-Bridge terdiri dari 4 buah transistor NPN. Basis pada transistor ini terhubung dengan keluaran dari gerbang AND sehingga aktif atau tidaknya transistor akan tergantung pada keluaran dari gerbang AND.

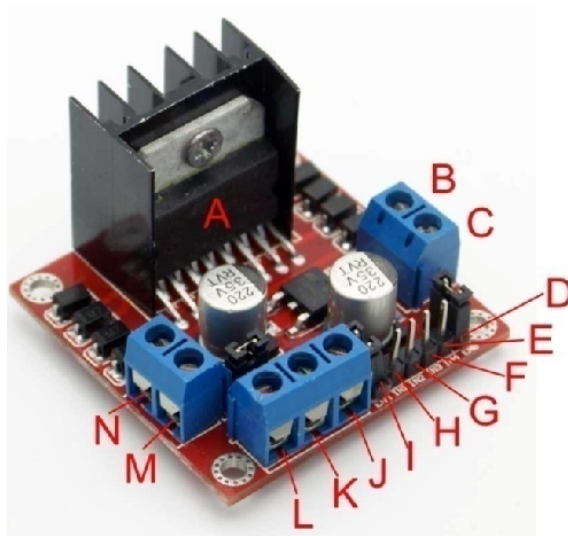
Keluaran dari modul driver motor ini akan bergantung pada pin In1, In2, In3, In4, EnA dan EnB pada driver motor L298. Keluaran pin In1, In2 dan EnA merupakan pin kontrol pada keluaran Out1 dan Out2. In1 dan In2 adalah pin yang digunakan untuk mengontrol arah arus yang keluar dari modul L298 ini dan pin EnA digunakan untuk mengontrol besarnya arus yang keluar dari driver motor ini. Sebagai contoh apabila pin In1 bernilai "1" atau mempunyai tegangan 5 volt dan pin In2 bernilai "0" atau mempunyai tegangan 0 Volt serta EnA mempunyai tegangan 5 Volt maka pin In1 dan EnA akan mengaktifkan gerbang AND1 di gambar 2.10 karena pada gerbang AND apabila kedua masukan bernilai 1 atau 5 Volt maka akan bernilai benar sehingga gerbang AND ini akan mengeluarkan keluaran bernilai 1 atau mempunyai tegangan 5 Volt. Tegangan 5 Volt ini akan mengaktifkan transistor yang berada pada keluaran gerbang AND menjadi saturasi sehingga transistor ini dapat mengeluarkan arus listrik. Namun pada gerbang AND2 keluaran bernilai 0 atau mempunyai tegangan 0 Volt karena pada masukan gerbang AND2 terdapat inverter yang digambarkan sebagai sebuah lingkaran. Inverter ini mengubah nilai In1 yang pada awalnya bernilai 1 menjadi 0 sehingga masukan pada gerbang AND2 ini menjadi 0 pada masukan dari In1 dan bernilai 1 pada masukan dari EnA oleh karena itu pada gerbang AND2 bernilai 0 karena gerbang AND hanya akan mengeluarkan keluaran 1 bila masing masing input mempunyai nilai 1. Arus listrik ini akan dikeluarkan pada pin OUT1. Selanjutnya pada In2 yang mempunyai nilai "0" atau mempunyai tegangan 0 Volt akan berpengaruh pada gerbang AND3 dan AND4. Pada gerbang AND3 masukan pertama dari IN2 yang bernilai 0 dan masukan kedua dari EnA yang bernilai 1. Gerbang AND akan mengeluarkan nilai 0 karena gerbang AND hanya akan bernilai 1 jika kedua masukan bernilai 1 sehingga transistor pada keluaran gerbang AND3 akan tidak aktif. Selanjutnya pada gerbang AND4 masukan dari In2 terdapat inverter sehingga masukan In2 dari bernilai 0 menjadi bernilai 1 dan masukan dari EnA tetap bernilai 1. Masukan dari In2 dan EnA yang sama-sama bernilai 1 akan menyebabkan gerbang AND4 bernilai 1 sehingga menyebabkan transistor pada keluaran gerbang

AND4 akan saturasi. Transistor dalam keadaan saturasi ini akan dapat mengalirkan arus listrik dari kolektor menuju emitter. Namun kolektor pad transistor ini tersambung dengan pin Out2 yang keadaannya sedang tidak tersambung pada apapun sehingga pin Out2 ini apabila diberi sumber tegangan maka akan segera menyerap arus listrik. Pin Out1 dan pin Out2 apabila diberi beban berupa motor maka akan ada arus listrik yang mengalir. Arus listrik yang mengalir pin Out1 dan pin Out2 akan dapat menggerakkan motor.

Pengendalian motor yang ke 2 mempunyai prinsip yang sama dengan pengendalian motor 1. Namun pin yang digunakan berbeda yaitu pin in1,in2 dan EnA diganti dengan pin in3,in4 dan EnB.



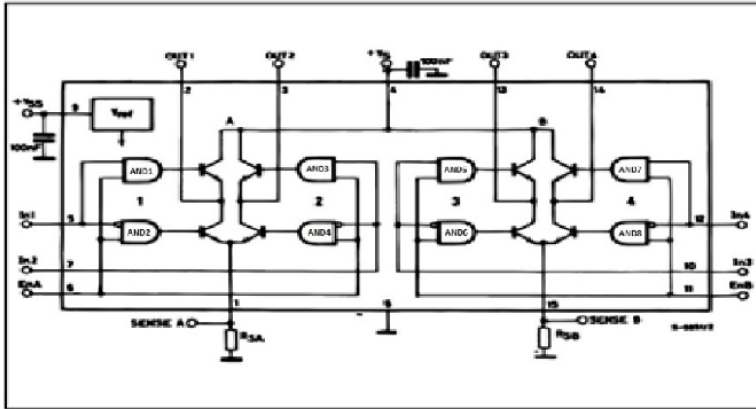
Gambar 2.8 Gambar Skematik Modul L298



Gambar 2.9 Pin Masukan dan Keluaran pada Modul L298

Keterangan gambar 2.8 dan gambar 2.9 :

- A : Integrated circuit L298
- B : pin Out 3 (keluaran untuk motor 2)
- C : pin Out 4 (keluaran untuk motor 2)
- D : pin EnB (masukan PWM untuk motor 2)
- E : pin In4 (masukan TTL untuk motor 2)
- F : pin In3 (masukan TTL untuk motor 2)
- G : pin In2 (masukan TTL untuk motor 1)
- H : pin In1 (masukan TTL untuk motor 1)
- I : pin EnA (masukan PWM untuk motor 1)
- J : pin keluaran 5 Volt
- K : pin *ground*
- L : pin masukan suplai daya maksimum 46 Volt
- M : pin Out 2 (keluaran untuk motor 1)
- N : pin Out 1 (keluaran untuk motor 1)



Gambar 2.10 L298 Modul Driver Motor

2.9. RF Transmitter dan Receiver 433 Mhz

Penerima dan pengirim frekuensi radio ini berfungsi pada frekuensi 433Mhz. Modul ini dapat dengan udah dimasukkan ke dalam *breadboard* dan berfungsi baik pada mikrokontroler untuk membuat *wireless data link* secara sederhana. Penerima dan pengirim ini bekerja secara satu arah tapi dapat dibuat dua arah dengan cara menambahkan sepasang lagi pengirim dan penerima. Modul ini dalam berkomunikasi mempunyai banyak gangguan seperti noise. Kedua pengirim dan penerima mempunyai frekuensi yang sama dan tidak mempunyai identitas atau alamat.

RF transmitter dan receiver 433 Mhz menggunakan modulasi ASK (Amplitude Shift Keying). Modulasi ASK merupakan modulasi yang pada awalnya digunakan dalam komunikasi radio telegraph diawal abad ke 19. Modulasi ASK adalah modulasi yang paling sederhana untuk di implementasikan namun sangat rentan terhadap *noise* dan distorsi[7]. Modulasi ASK adalah modulasi yang mengubah-ubah amplitude sesuai dengan adanya sinyal yang akan dikirimkan.

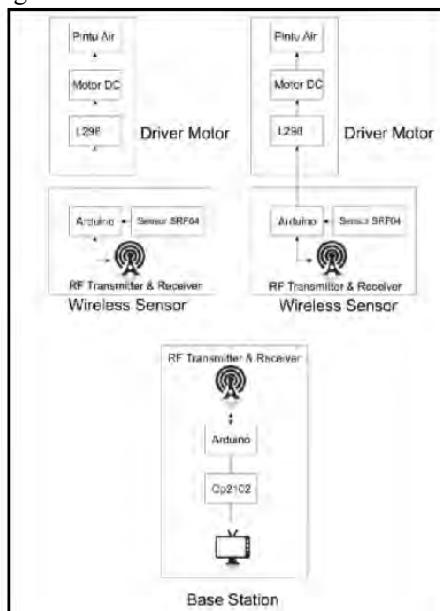


Gambar 2.11 Modul Pengirim (kiri) dan Penerima (kanan) RF 433Mhz

BAB III

PERANCANGAN SISTEM

Pembuatan tugas akhir ini ada dua system utama yang dibangun yaitu *base station* dan *node wireless sensor*. *Base station* berfungsi untuk mengumpulkan data dari masing-masing *node* kemudian mengolah data yang diterima selanjutnya memproses data dengan menggunakan On/Off lalu mengirimkan hasil proses data menuju *node wireless sensor*. Selain itu juga terhubung pada komputer untuk mengirim data ke komputer agar dapat ditampilkan pada layar komputer. Sedangkan pada *node wireless sensor* bertugas untuk mengukur ketinggian air, mengirim data ke base station, menerima dan melakukan perintah dari *base station*. Sistem keseluruhan dapat diilustrasikan pada gambar 3.1



Gambar 3.1 **Diagram Blok Sistem**

Ilustrasi pada gambar 3.1 dapat dilihat pada *base station* terdapat komputer, arduino dan modul komunikasi *radio frequency FS1000a 433 Mhz* dan *XY-MK-05*. Sedangkan pada *node wireless sensor* terdapat

arduino, sensor HC-SR04, pengendali motor DC L298, motor DC dan pintu air.

3.1. Perancangan *Base Station*

Perangkat keras yang digunakan dalam *base station* adalah arduino uno, modul RF *transmitter* FS1000A dan *receiver* XY-MK-5V, USB to serial converter CP2102 dan komputer serta perangkat lunak Delphi 7 serta perangkat lunak arduino.

3.1.1 Perancangan Perangkat Keras *Base Station*

Arduino Uno pada *base station* digunakan untuk memberikan perintah pada *node* untuk memberikan data hasil pembacaan sensor setelah itu mengolah data yang di dapat dengan menggunakan kontrol On/Off lalu data yang diolah kembali dikirimkan menuju *node* untuk di aplikasikan pada pengatur motor berupa motor DC.

Gambar 3.2 *Transmitter* FS1000A yang mempunyai frekuensi 433Mhz digunakan untuk mengirim data sedangkan *receiver* XY-MK-5V digunakan untuk menerima data yang telah dikirim. *Transmitter* FS1000A dihubungkan dengan tegangan 5V agar dapat memancarkan data. Data dari *transmitter* FS1000A di hubungkan dengan pin 12 arduino. Sedangkan pada *receivernya* di hubungkan dengan pin 11 arduino. Suplai daya pada *base station* diberikan oleh komputer sebesar 5V dari tegangan USB.

Komputer pada *base station* digunakan untuk menampilkan hasil ketinggian air yang dibaca oleh *node-node wireless sensor*. Tampilan dari *base station* di program dengan menggunakan program Delphi yang ditampilkan dalam bentuk grafik.



Gambar 3.2 Diagram *Base Station*



Gambar 3.3 Perangkat Keras *Base Station*

3.1.2 Diagram Alir Kerja *Base Station*

Diagram alir kerja *base station* ditunjukkan pada gambar 3.4. Saat pertama kali menyala *base station* akan mengambil data yang ada pada *node-node*. Namun data dari *node-node* ini tidak bisa langsung di ambil bersamaan sehingga dibutuhkan pengambilan data secara bergantian.

Pertama pengambilan data di lakukan pada *node wireless sensor* yang mempunyai alamat A dengan cara mengirim perintah beserta alamat dalam 1 paket. Selanjutnya *base station* akan menunggu *node A* membalas. Setelah dibalas dan data diterima dari *node A* maka akan

disimpan. Kedua pengambilan data di lakukan juga pada *node* B dengan cara yang sama dilakukan pada *node* A namun alamat diubah menjadi alamat *node* B.

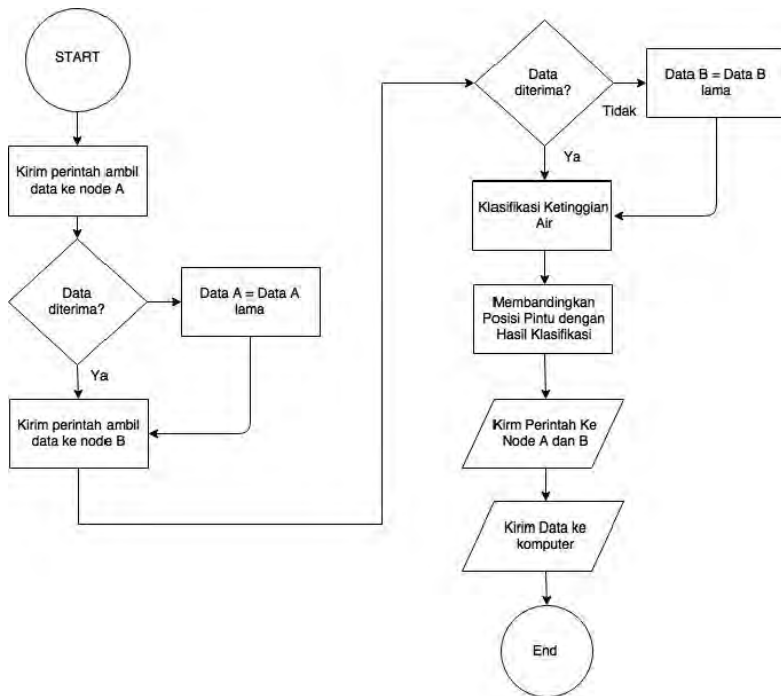
Perintah yang dikirimkan kepada *node* A dan *node* B berupa sebuah paket data yang mempunyai susunan dalam sebuah *frame* pengiriman {perintah, alamat, alamat, data1, data2, data3}. Perintah diwakili oleh beberapa kode yaitu kode “U” untuk meminta data hasil pembacaan ultrasonik pada *node*, kode “W” merupakan jawaban dari *node* mengenai data pembacaan ultrasonik, kode “F” digunakan untuk memberi perintah pada *node* agar mengimplementasikan hasil kontrol On/Off pada *driver* motor.

Semua data dari *node wireless sensor* didapat maka akan diproses dalam kontrol On/Off. Kontrol On/Off pada *base station* ini akan dapat memberi perintah pada *node* agar membuka ataupun menutup pintu air pada posisi tertentu sesuai dengan hasil pengklasifikasian ketinggian air. Perintah membuka menutup pintu air akan diberikan apabila ketinggian air melebihi atau kurang dari ketinggian air yang ditentukan.

Perintah membuka atau menutup pintu air ini dikirimkan kembali pada *node* A dan *node* B. Perintah ini diimplementasikan pada *driver* motor oleh *node* untuk memutar motor agar dapat membuka dan menutup pintu air. Naik dan turunnya pintu air ini di tentukan dengan arah putaran motor DC.

Pengiriman data dari *base station* menuju Delphi adalah dengan menggunakan 6 paket data yaitu 3 data pertama adalah paket data untuk ketinggian air pada pintu air 1 dan paket data kedua adalah ketinggian air pada pintu air kedua. 6 paket data ini akan dipisah menjadi 2 yaitu 3 paket data pertama dan 3 paket data kedua. Dari 3 paket data ini akan dimasukkan kedalam TChart pada program Delphi agar bisa tampil mereperesntasikan grafik ketinggian air. Pada sumbu x dari grafik adalah waktu per detik sampling data yang diambil dari *base station*.

Gambar 3.4 merupakan diagram alir pada *base station*. Diagram alir ini menunjukkan aktifitas yang dilakukan oleh *base station*.



Gambar 3.4 Diagram Alir *Base Station*

3.1.3 Perancangan Perangkat Lunak *Base Station*

Perangkat lunak pada *base station* ini dijalankan oleh mikrokontroler yang ada pada arduino untuk mengendalikan seluruh algoritma. *Base station* menggunakan bahasa pemrograman C pada lingkungan arduino. Pada program ini terdapat *library virtual wire* untuk pengiriman dan penerimaan data, pengiriman data ke komputer serta kontroler On/Off.

3.1.3.1. Program *Virtual Wire*

Virtual wire digunakan untuk membuat *transmitter* dan *receiver* agar dapat berkomunikasi satu sama lain. Kecepatan transmisi pada *virtual wire* di set pada kecepatan 2000 *bit per second*.


```

Program inisialisasi virtual wire
#include <VirtualWire.h>
const int transmit_pin = 12;
const int receive_pin = 11;
Selanjutnya penentuan pin yang digunakan untuk transmitter dan
receiver dan kecepatan komunikasi data:
vw_set_tx_pin(transmit_pin);
vw_set_rx_pin(receive_pin);
vw_setup(2000);
vw_rx_start();
Berikut ini program untuk pengiriman data antara transmitter dan
receiver
char msgAA[7] = {'U','A',' ',' ',' ','I','#'};
vw_send((uint8_t *)msgAA, 7);
vw_wait_tx();
dimana char msgAA adalah array yang berisi 7 buah tipe data char
yang akan dikirim. Pengiriman dilakukan dengan perintah
vw_send((uint8_t *)msgAA, 7) lalu pengiriman ditunggu hingga semua
data terkirim dengan perintah vw_wait_tx(). Pada bagian terakhir adalah
penerimaan data. Penerimaan data dilakukan cara menunggu adanya
data yang masuk selama maksimal 1 detik, jika ada yang masuk maka
data tersebut akan segera diproses dengan program sebagai berikut:
vw_wait_rx_max(1000);
if (vw_get_message(bufAA, &buflenAA))
{
ketinggian_airAA = bufAA[2]*100 + bufAA[3]*10 + bufAA[4];
}

```

3.1.3.2. Program Pengiriman Data Ke Komputer

Pengiriman data ke komputer dilakukan agar dapat menampilkan data dalam bentuk grafik pada komputer. Data dikirim dalam bentuk barisan data yang terdiri dari 6 buah data angka. Tiga digit pertama adalah data untuk ketinggian air pada pintu air pertama dan 3 digit selanjutnya adalah untuk ketinggian air pada pintu air yang kedua.

```

if (ketinggian_airAA < 100)
{
    if (ketinggian_airAA > 10)

```

```

    {
    Serial.print(0);
    Serial.print(ketinggian_airAA);
    }
}
if (ketinggian_airAA < 10)
{
    Serial.print(0);
    Serial.print(0);
    Serial.print(ketinggian_airAA);
}
if (ketinggian_airAA > 100)
{
    Serial.print(ketinggian_airAA);
}
digitalWrite(led_pin, LOW);
delay(10);

if (ketinggian_airBB < 100)
{
    if (ketinggian_airBB > 10)
    {
        Serial.print(0);
        Serial.print(ketinggian_airBB);
    }
}
if (ketinggian_airBB < 10)
{
    Serial.print(0);
    Serial.print(0);
    Serial.print(ketinggian_airBB);
}
if (ketinggian_airBB > 100)
{
    Serial.print(ketinggian_airBB);
}
}

```

Program pengiriman data ke komputer ini pertama membagi nilai ketinggian air pada pintu A dan B pada nilai-nilai tertentu. Jika ketinggian air A dan B berada diantara nilai 11-99 maka akan diberi

angka 0 di depannya agar nilai menjadi 6 digit. Sebagai contoh nilai ketinggian air A dan B 50 dan 75 maka akan ditambahkan nilai 0 di depannya menjadi 050 dan 075. Jika ketinggian air A dan B berapa diantara 0-9 maka akan di tambahkan 2 buah angka 0 di depannya misalkan nilai ketinggian A adalah 6 dan ketinggian B adalah 8 maka akan diubah menjadi 006 dan 008 sehingga data yang terkirim ke komputer adalah 006008.

3.1.3.3. Program Kontroler On/Off

Kontrol On/Off yang digunakan dalam mengatur pintu air ini menggunakan cara apabila ketinggian air yang ada kurang ataupun melebihi yang diinginkan maka pintu air akan tertutup atau terbuka. Pada kontrol ini ketinggian air yang telah dibaca akan diklasifikasikan menurut ketinggiannya.

Tabel 3.1 Klasifikasi Ketinggian Air

Ketinggian Air Pintu 1 (mm)	Klasifikasi Ketinggian Air	Ketinggian Air Pintu 2 (mm)	Klasifikasi Ketinggian Air
<= 160	0	<= 140	0
161 - 170	1	141-150	1
171 - 180	2	151-160	2
181 - 190	3	161-170	3
191 - 200	4	171-180	4
201 - 210	5	181-190	5

Tabel 3.1 menunjukkan konversi yang dari ketinggian air pada masing masing pintu menjadi sebuah angka hasil klasifikasi ketinggian pintu air. Pada program selanjutnya setelah pengklasifikasian selesai maka akan dibandingkan dengan nilai posisi pintu air. Pada saat awal posisi pintu air diset pada posisi 0 yaitu posisi tertutup sehingga bernilai 0. Nilai posisi pintu air dan klasifikasi pintu air ini lah yang dibandingkan untuk mendapatkan sebuah kontrol pintu air terbuka atau tertutup. Nilai klasifikasi menunjukkan bukaan pintu air dari yang bernilai 0 yaitu tertutup hingga bernilai 5 yaitu terbuka keseluruhan. Di tabel 3.2 menunjukkan lebar bukaan pintu air yang mempunyai nilai antara 0 hingga 5. Jika nilai klasifikasi pintu air lebih besar dari posisi pintu air maka pintu air akan terbuka apabila sama maka pintu air akan

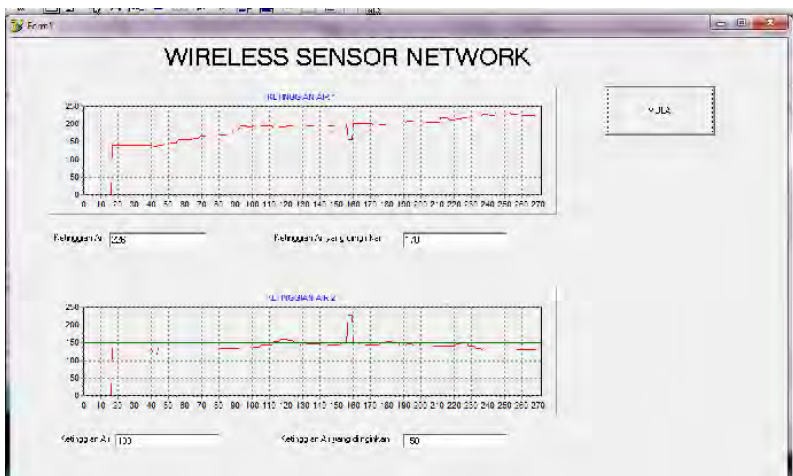
diam dan apabila lebih kecil maka akan menutup. Sehingga posisi pintu akan menyesuaikan ketinggian air yang terjadi.

Tabel 3.2 Lebar Buka-an Pintu Air

Klasifikasi Ketinggian Air	Lebar Buka-an Pintu Air (mm)
0	0
1	4
2	8
3	12
4	16
5	20

3.1.3.4. Program Delphi

Program Delphi digunakan untuk menampilkan data ketinggian air sungai berupa grafik dan angka. Hal ini untuk memudahkan untuk melihat perubahan ketinggian air sungai dari waktu ke waktu.



Gambar 3.5 Grafik Penampil Data pada Delphi

Dari Gambar 3.5 dapat dilihat bahwa terdapat grafik ketinggian pada sungai 1 dan sungai 2. Pada grafik ketinggian air 1 terlihat

terdapat 2 garis yaitu berwarna merah dan berwarna hijau. Garis berwarna merah menunjukkan ketinggian air yang dibaca sedangkan garis berwarna hijau menunjukkan ketinggian air yang diinginkan. Pada ketinggian air 2 juga mempunyai 2 grafik yaitu berwarna merah dan berwarna hijau. Grafik berwarna merah menunjukkan ketinggian air yang terbaca dan yang berwarna hijau menunjukkan ketinggian yang diinginkan.

3.2. Perancangan Node Wireless Sensor

Perangkat keras yang digunakan dalam *node wireless sensor* adalah arduino uno, modul RF *transmitter* FS1000A dan *receiver* XY-MK-5V serta sensor ultrasonik .

3.2.1 Perancangan Perangkat Keras Node Wireless Sensor

Arduino uno dalam *node wireless sensor* bertugas untuk mengatur seluruh kegiatan yang dilakukan oleh *node* seperti berkomunikasi, mengukur ketinggian dengan sensor ultrasonik HC-SR04 dan mengimplementasikan hasil pengolahan kontroler On/Off dari *base station* menuju *driver motor* L298.

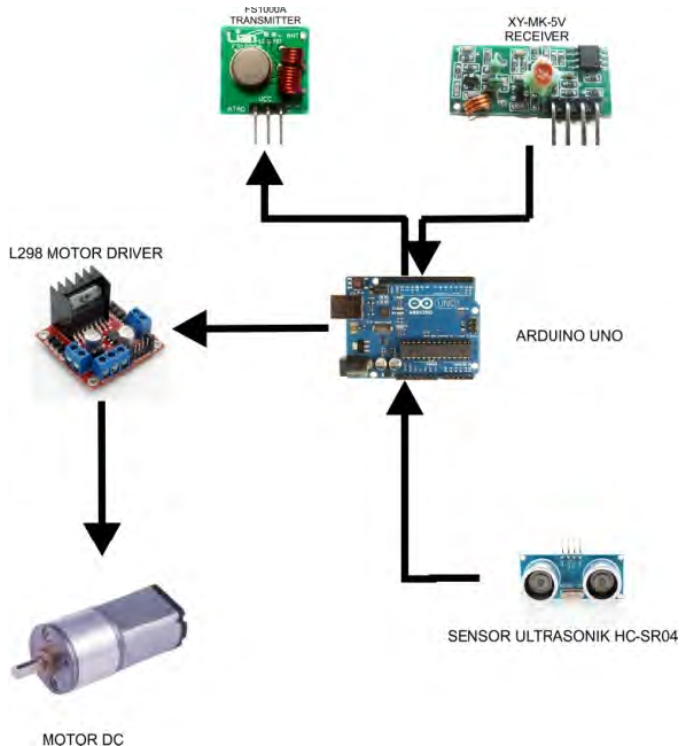
Komunikasi yang dilakukan oleh *node* menggunakan perangkat keras yang sama dengan *base station* yaitu RF *transmitter* FS1000A dan *receiver* XY-MK-5V. Ketika saling berkomunikasi *transmitter* pada *base station* akan mengeluarkan sinyal dan diterima oleh *receiver* pada *node*. *Node* akan membalas dengan mengeluarkan sinyal pada *transmitter*nya dan diterima oleh *base station* dengan menggunakan *receiver*nya. Sama halnya dengan *base station*, *transmitter* juga dihubungkan pada tegangan 5 Volt dan data keluaran dari *transmitter* dihubungkan dengan pin 11 sedangkan *receiver*nya pada pin 12.

Node mengambil data ketinggian menggunakan sensor ultrasonik HC-SR04. Sensor ultrasonik mempunyai 4 pin yaitu VCC, *trigger*, *echo*, *ground*. Pin VCC dan *ground* masing masing dihubungkan pada 5V dan *ground* pada arduino. Sedangkan pin *trigger* dihubungkan dengan pin 7 dan pin *echo* dihubungkan dengan pin 8.

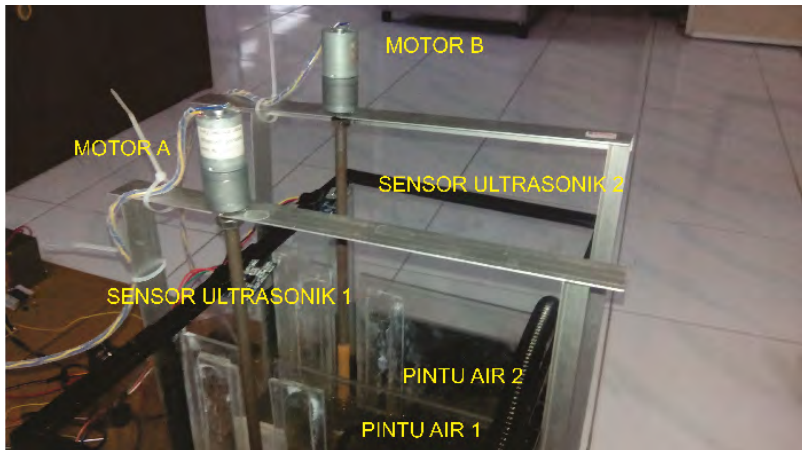
Selanjutnya *node* akan menunggu balasan dari *base station* berupa nilai hasil kontroler On/Off yang dilakukan di *base station*. Hasil perhitungan yang diterima akan diubah ke sinyal *pwm* untuk dikirimkan

menuju L298 dan juga dari hasil perhitungan akan menentukan arah putaran motor.

Driver motor L298 digunakan untuk mengubah sinyal *pwm* menjadi gerakan perputaran motor. Kecepatan motor diatur oleh besar kecilnya *duty cycle* dari *pwm* dan arah perputaran motor ditentukan oleh pin digital. Sinyal *pwm* diambil dari pin 6 arduino dan untuk sinyal digital diambil dari pin 4 dan 2. Sedangkan pada *driver motor* L298 pin 4 dan 2 arduino dihubungkan pada in1 dan in2. Dan *pwm* dari arduino dihubungkan dengan EnA.



Gambar 3.6 Diagram *Node Wireless Sensor*



Gambar 3.7 Peletakan Aktuator dan Sensor

3.2.2 Diagram Alir Wireless Sensor

Node wireless sensor pada saat menyala selalu pada mode pasif yaitu mode dimana dia menunggu perintah dari *base station* untuk melakukan sesuatu. Pada saat *node* menerima sebuah paket data maka paket data tersebut dibaca.

Node ini adalah sebuah titik sistem yang dapat ditempatkan pada tempat tertentu untuk dapat mengukur besaran tertentu. Besaran yang diukur pada *node* ini adalah besaran ketinggian air sungai.

Aktivitas yang dapat dilakukan *node* adalah membaca sensor lalu mengirim data pada *base station* dan mengaktualisasikan hasil proses data dengan kontroler On/Off yang dilakukan oleh *base station*.

Ketika *node* diberi perintah untuk mengambil data dari sensor dan mengirimkan hasil pembacaan sensor kembali ke *base station* maka yang dilakukan *node* adalah mengambil data dari sensor ultrasonik dan mengirimkannya kembali menuju *base station*. Setelah itu ketika *node* diberi perintah untuk mengaktualisasikan hasil perhitungan kontroler pada *base station*.

Node wireless sensor mempunyai 2 mode yaitu mode untuk membaca sensor serta mode untuk mengimplementasikan kontroler

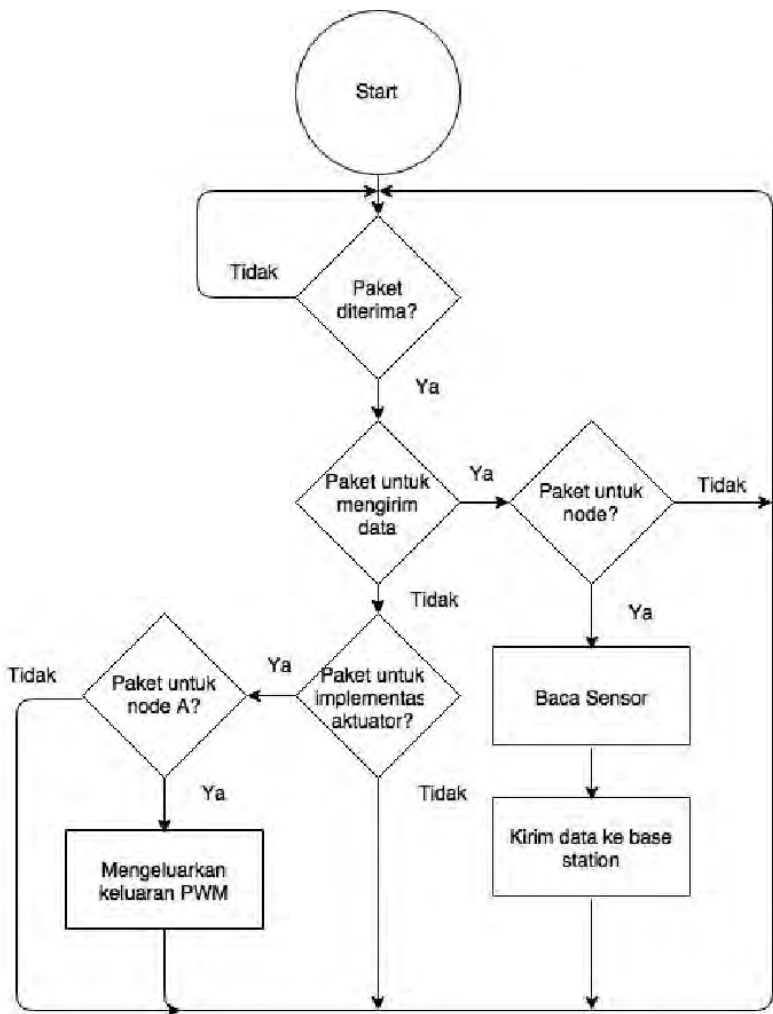
On/Off pada motor. Mode pembacaan ketinggian air dengan menggunakan ultrasonic.

Node *wireless sensor* akan menjadi aktif ketika adanya perintah dari *base station* dengan awalan header nya adalah “U”. Header ini menunjukkan bahwa *base station* menginginkan ketinggian yang ada pada sebuah *node*. Ketika header ini di dapat oleh *node* maka akan langsung membaca ketinggian air dengan sensor ultrasonik lalu mengirimkannya kembali pada *base station* dengan header awalan “W”. Jadi ketika masukan berupa header “U” maka *wireless sensor* ini akan mengirimkan header “W” . Header tersebut menunjukkan jawaban dari header yang dikirimkan oleh *base station* yaitu “U”.

Mode yang kedua yaitu mode untuk mengimplementasikan kontroler On/Off pada motor. Pada mode ini *node wireless sensor* menunggu perintah untuk mengimplementasikan dengan header awalan berupa “F”. Header ini berarti perintah untuk mengimplementasikan kontrol On/Off untuk segera di implementasikan pada motor DC.

Pengimplementasian pada motor DC ini akan memberikan sinyal *pulse width modulation* sebesar 255 atau keluaran tegangan yang ada sebesar 5 volt apabila *base station* memerintah *node* agar dapat menjadikan motor menjadi bergerak atau pada posisi On. Namun apabila *base station* diberikan perintah untuk menjadikan motor berhenti bergerak maka akan mengirim nilai *pulse width modulation* sebesar 0 atau motor dalam kondisi Off.

Sehingga mode *node wireless sensor* merubah data yang dikirimkan *base station* menjadi sinyal *pwm* dan sinyal untuk menentukan arah putaran motor DC yang nantinya berpengaruh pada naik turunnya pintu air.



Gambar 3.8 Diagram Alir *Node Wireless Sensor*

3.2.3 Perancangan Perangkat Lunak *Wireless Sensor Node*

Perangkat lunak pada *wireless sensor* ini dijalankan oleh mikrokontroler yang ada pada arduino untuk mengendalikan seluruh algoritma. *Wireless sensor* menggunakan bahasa pemrograman C pada lingkungan arduino. Pada program ini terdapat *library virtual wire* untuk pengiriman dan penerimaan data, pembacaan nilai sensor ultrasonik, serta *pulse width modulation* untuk mengatur pergerakan motor.

3.2.3.1. Pembacaan Sensor Ultrasonik

Pembacaan sensor ultrasonik dilakukan dengan memberikan sinyal *trigger* selama 10 mikro sekon lalu dihitung lama dari sinyal *echo* kondisi *high*. Lama sinyal pin *echo* dalam kondisi *high* akan dihitung dengan rumus tertentu agar mendapatkan jarak pengukuran sensor ultrasonik.

```
digitalWrite(trigPin, LOW);  
delayMicroseconds(2);  
digitalWrite(trigPin, HIGH);  
delayMicroseconds(10)  
digitalWrite(trigPin, LOW);  
duration = pulseIn(echoPin, HIGH);  
distance = 250 - ((duration/2)*10) / 22;
```

Pembacaan ini fungsi `digitalWrite()` akan memberikan nilai 1 atau 0. Sesuai dengan timing diagram sensor ultrasonik HC-SR04 arduino memberikan nilai *low* pada pin trigger selama 2 mikrosekon. Setelah itu memberikan nilai *high* selama 10 mikrosekon untuk mengaktifkan sinyal burst pada modul ultrasonik ketika nilai pin *trigger* dari *high* berubah menjadi *low*. Setelah itu sinyal burst telah dikeluarkan maka pin *echo* akan mempunyai kondisi *high* hingga sinyal burst diterima kembali oleh sensor ultrasonik. Pembacaan nilai *echo* akan dihitung oleh perintah `pulseIn()` ketika pin *echo* pada kondisi *high*. Hasil pembacaan nilai `pulseIn()` akan dimasukkan dalam variabel `duration`. Variabel `distance` adalah pengolahan variabel `distance` agar menjadi sebuah jarak yang sebenarnya dengan rumus $distance = 250 - ((duration/2)*10) / 22$. Nilai `duration/2` didapat karena perjalanan dari gelombang ultrasonik mengalami 2 kali perjalanan yaitu ketika keluar dari ultrasonik dan setelah terpantul dari objek hingga kembali ke sensor ultrasonik.

3.2.3.2. Implementasi Aktuator di Driver Motor L298

Node akan menerima perintah dari base station untuk mengatur arah dan kecepatan dari motor yang diatur. Nilai yang akan diterima oleh *node* untuk mengatur motor bernilai antara 0-255 dimana 0-127 menyatakan bahwa pintu air akan bergerak turun dan kecepatannya berdasarkan nilai antara 0-127. Dimana semakin besar nilainya maka semakin besar pula kecepatan untuk naik turunnya pintu air dengan nilai tersebut dikalikan 2. Kecepatan naik turunnya pintu air ini diatur oleh sinyal *pwm* yang keluar dari pin 6 arduino. Pergerakan turunnya pintu air ditentukan oleh pin 4 dalam keadaan *high* dan pin 2 dalam keadaan *low*. Sedangkan untuk nilai antara 128-255 adalah nilai untuk menaikkan pintu air. Pergerakan naiknya pintu air akan ditandai dengan pin 4 bernilai *low* dan pin 2 bernilai *high*. Nilai *pwm* yang dihasilkan dengan cara nilai 128 hingga 255 akan dikurangkan dengan 128 untuk menghasilkan nilai 0-127 lalu dikalikan dengan 2 untuk mendapatkan hasil *pwm*. Berikut program untuk mengendalikan driver motor L298.

```
Aktuator_A = buf[3]*100 + buf[4]*10 + buf[5];
```

```
Serial.print("buf=");  
Serial.print(buf[3]);  
Serial.print(buf[4]);  
Serial.print(buf[5]);  
Serial.print("=");
```

```
Serial.print("AKTUATOR A =");  
Serial.print(Aktuator_A);  
Serial.print("=");  
if(Aktuator_A > 127)
```

```
{  
  digitalWrite(4,HIGH); // PPINTU AIR NAIK  
  digitalWrite(2,LOW);
```

```
Aktuator_AS = ((Aktuator_A - 127) ); //Aktuator_A mulai  
dari 0-255
```

```
if (Aktuator_AS > 255)  
{  
  Aktuator_AS = 255;
```

```

}
if (Aktuator_AS < 0)
{
Aktuator_AS = 0;
}

analogWrite(6,Aktuator_AS);
Serial.print("NAIK=");
Serial.print(Aktuator_AS);
Serial.print("=");
}

```

```

if(Aktuator_A < 126)
{
digitalWrite(2,HIGH); // PINTU AIR TURUN
digitalWrite(4,LOW);
Aktuator_AS = ((126 - Aktuator_A) ); //Aktuator_A mulai
dari 0-255

```

```

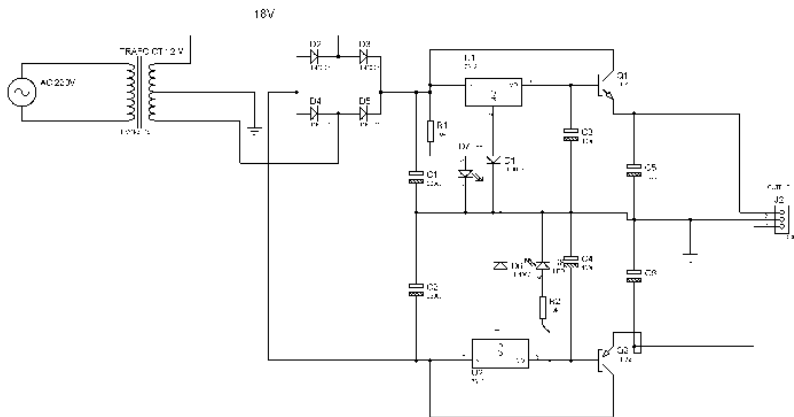
if (Aktuator_AS > 255)
{
Aktuator_AS = 255;
}
if (Aktuator_AS < 0)
{
Aktuator_AS = 0;
}

analogWrite(6,Aktuator_AS);
Serial.print("TURUN=");
Serial.print(Aktuator_AS);
Serial.print("=");
}

```

3.3. Perancangan Suplai Daya

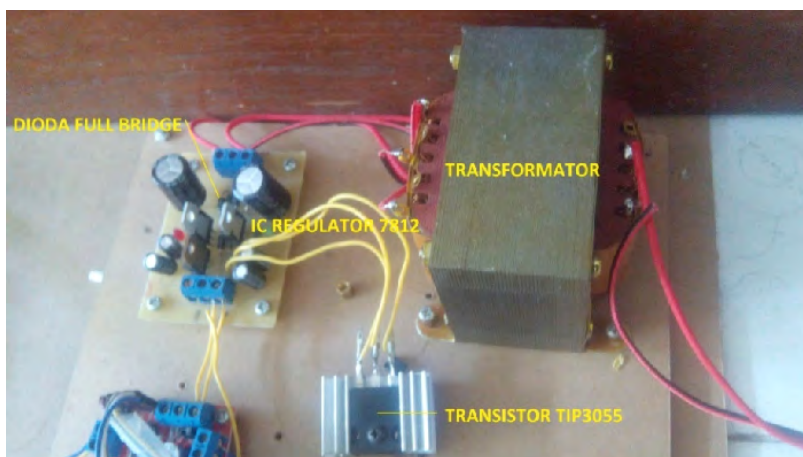
Suplai daya digunakan untuk memberikan daya ke *node wireless sensor*, driver motor L298 dan motor DC. Suplai daya yang dirancang ini mempunyai tegangan yang diinginkan sebesar 12V. Pada gambar 3.9 rangkaian menggunakan transformator *centertap*, *diode full wave*, regulator 12 Volt, transistor arus TIP3055 dan kapasitor sebagai filter.



Gambar 3.9 Skema Rangkaian Suplai Daya

Transformator digunakan untuk menurunkan tegangan dari tegangan 220 Volt menjadi tegangan 18 Volt. Tegangan 18 Volt yang dihasilkan oleh transformator masih dalam bentuk sinyal AC sehingga memerlukan penyearah gelombang penuh untuk menjadikan sinyal AC menjadi sinyal DC. Penyearah gelombang penuh terdiri dari 4 buah diode 1N4002. Setelah gelombang menjadi searah atau DC maka tegangan DC 18V distabilkan dengan menggunakan IC regulator 7812 yang dapat merubah tegangan 18V DC menjadi 12V DC. IC regulator mempunyai input tegangan sebesar 35 Volt sehingga tegangan 18 Volt masih dapat dijangkau oleh regulator ini namun regulator ini mempunyai kekurangan yaitu arus yang dihasilkan sebesar 1 Ampere. Oleh karena itu diperlukan rangkaian untuk memperkuat arus yang dikeluarkan oleh IC regulator 7812 yaitu transistor TIP3055 pada gambar 3.9 ditunjukkan dengan nama Q1. Dioda yang berada pada pin 2

pada IC regulator 7812 meningkatkan keluaran tegangan 7812 yang seharusnya 12 Volt menjadi 12.7 Volt dikarenakan tegangan yang terjadi pada diode D1 ketika keadaan forward bias menjadi 0.7. Tegangan 12.7 ini Keluaran 7812 ini disambungkan pada basis transistor TIP3055 sehingga 7812 akan mensuplai arus basis pada transistor . Transistor ini mempunyai keluaran arus hingga 15 Ampere. Gambar 3.10 menunjukkan penempatan dari komponen-komponen suplai daya ini



Gambar 3.10 Suplai Daya

Komponen yang lain adalah indikator led. Indikator ini mengindikasikan nyala atau tidaknya suplai daya ini. LED dihubungkan dengan tegangan keluaran dioda full wave dan terdapat resistor untuk memberikan arus yang dibutuhkan oleh LED agar tidak terjadi arus yang berlebihan.

3.4. Perancangan Prototip Pintu Air Sungai

Perancangan prototip sungai menggunakan akuarium yang mempunyai ukuran panjang 39.5 cm lebar 25 cm dan tinggi 27 cm. akuarium ini dibagi menjadi 3 bagian untuk membuat miniatur pintu air yang masing-masing pintu air mempunyai jarak 13 cm.

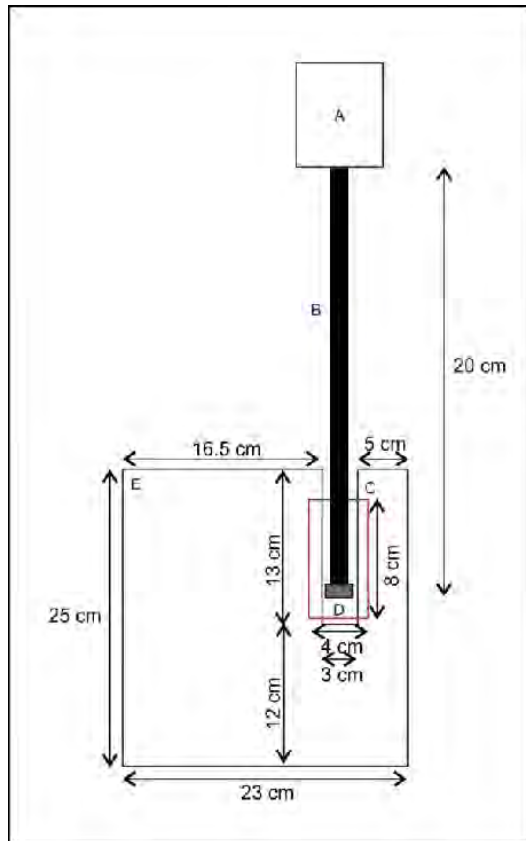
Pintu air menggunakan pintu air yang digerakkan oleh motor DC yang dapat menggerakkan pintu air keatas dan kebawah. Putaran motor DC ini menyebabkan pintu air dapat bergerak. Pada gambar 3.11 menunjukkan desain dari pintu air yang terdiri dari motor DC. Motor DC ini terkopel dengan baut sehingga ketika motor DC ini berputar maka bautpun juga ikut berputar. Putaran antara motor DC dan baut akan sama yaitu ketika motor DC berputar searah jarum jam maka baut akan berputar searah jarum jam dan apabila motor DC berputar berlawanan dengan arah jarum jam maka baut akan berputar berlawanan dengan arah jarum jam.

Selanjutnya untuk mengkonversi putaran dari baut ini maka menggunakan mur yang ditempel pada pintu air sehingga putaran dari baut akan dikonversi oleh mur menjadi gerakan linier. Gerakan linier ini dapat menggerakkan pintu air menjadi naik dan turun. Naik dan turunya pintu air ini menjadikan pintu air dapat terbuka dan tertutup.

Gambar 3.12 menunjukkan keseluruhan dari prototip yang terdiri dari 2 pintu air. Pintu air ini di pasang dengan jarak masing-masing sebesar 13 cm. Jarak antar pintu air ini akan membentuk ruang yang dapat menampung air. Ruang ini akan diatur ketinggian airnya oleh pintu air.

Prototip ini mempunyai pompa air yang menghubungkan bagian hulu dan hilir yang digunakan untuk mensimulasikan aliran air dari hulu ke hilir. Aliran air ini akan melewati pintu air untuk masuk ke hilir sungai. Bagian hulu dari prototip ini terletak di sebelum pintu air 1 sedangkan bagian hilir terletak di setelah pintu air 2.

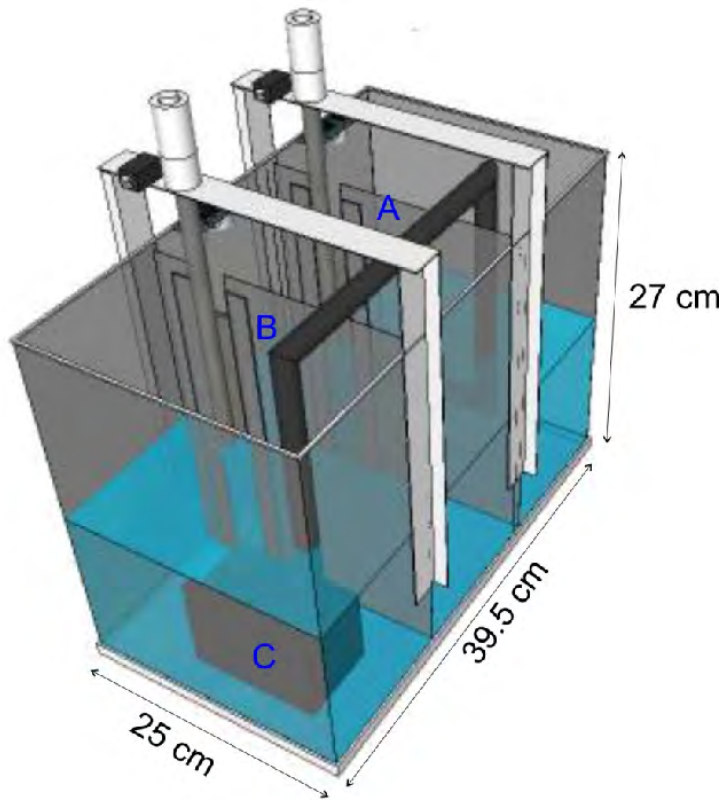
Pengukuran ketinggian air sungai ini diukur oleh sensor ultrasonik yang dipasang ditepian akuarium dengan ketinggian 25 cm. sensor ultrasonik dipasang sebelum pintu air 1 untuk mengukur ketinggian air pada pintu air 1 dan dipasang pada pintu air 2 untuk mengukur ketinggian air pada pintu 2.



Gambar 3.11 Desain Pintu Air

Keterangan

- A : Motor DC
- B : Baut
- C : Pintu Air
- D : Mur
- E : Bendungan



Gambar 3.12 Desain Pintu Air Keseluruhan

Keterangan

- A : Pintu Air 1
- B : Pintu Air 2
- C : Pompa Air

Halaman ini sengaja dikosongkan

BAB IV

PENGUJIAN SISTEM

Bab ini akan dijelaskan tentang pengujian sistem dimana hasil rancangan akan diuji dan dibahas. Pengujian sistem akan disertai hasil dan gambar yang mendukung.

4.1. Pengujian Sensor Ultrasonik HC-SR04

Pengujian ini bertujuan untuk mengetahui jarak yang diukur oleh sensor ultrasonik. Data hasil pengukuran sensor ultrasonik akan dibandingkan dengan pengukuran dengan penggaris dan dilihat perbedaan yang terjadi antara pengukuran dengan penggaris.

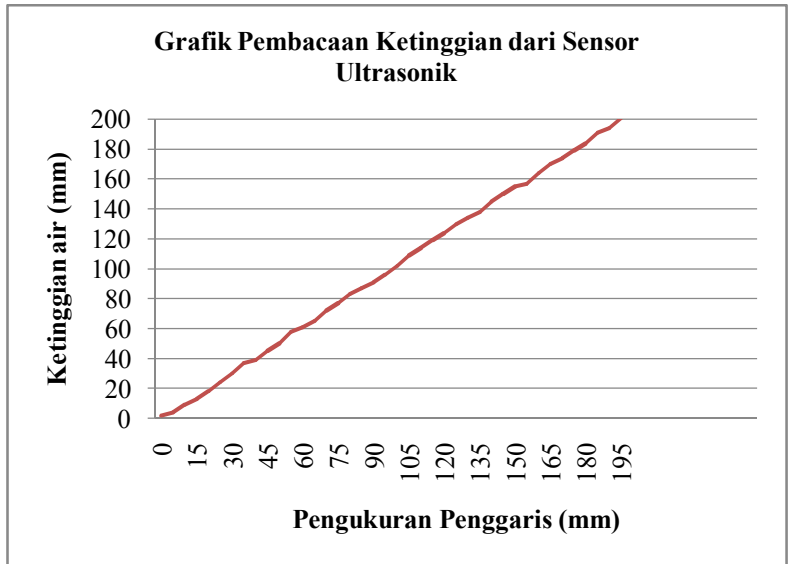
Terlihat pada gambar 4.2 pengukuran dilakukan dengan cara meletakkan sensor ultrasonik diatas air dan ditaruh pada ketinggian 250 mm.

Tabel 4.1 Pengujian Sensor HC-SR04

Nomor	Penggaris (mm)	Data Sensor SRF (mm)	Perbedaan (mm)
1	0	2	2
2	5	4	-1
3	10	9	-1
4	15	13	-2
5	20	18	-2
6	25	24	-1
7	30	30	0
8	35	37	2
9	40	39	-1
10	45	45	0
11	50	50	0
12	55	58	3
13	60	61	1
14	65	65	0
15	70	72	2

Tabel 4.1 Pengujian Sensor HC-SR04 (lanjutan)

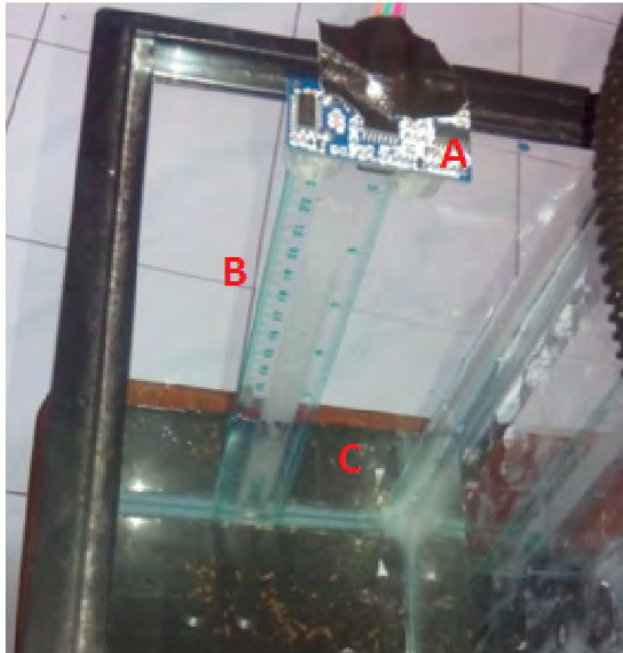
Nomor	Penggaris (mm)	Data Sensor SRF (mm)	Perbedaan (mm)
16	75	77	2
17	80	83	3
18	85	87	2
19	90	91	1
20	95	96	1
21	100	102	2
22	105	109	4
23	110	114	4
24	115	119	4
25	120	124	4
26	125	130	5
27	130	134	4
28	135	138	3
29	140	145	5
30	145	150	5
31	150	155	5
32	155	157	2
33	160	164	4
34	165	170	5
35	170	174	4
36	175	179	4
37	180	184	4
38	185	191	6
39	190	194	4
40	195	201	6



Gambar 4.1 Grafik Pengukuran Sensor Ultrasonik terhadap Ketinggian Air

Terlihat dalam tabel 4.1 dan grafik 4.1 bahwa pengukuran yang dilakukan oleh ultrasonik mengalami kesalahan semakin besar. Kesalahan Rata-rata keseluruhan dari pengukuran sebesar 2.375 mm. hal ini ber sesuaian dengan datasheet sensor ultrasonik dengan ketelitian mencapai 3 mm.

Pengujian sensor ultrasonik ini dengan cara meletakkan sensor ultrasonik berapada pada tepian dari akuarium. Selanjutnya diberikan penggaris sebagai referensi pengukuran. Lalu air diisi sedikit demi sedikit dengan menambah ketinggian sebesar 5 mm yang di sesuaikan dengan penggaris. Terakhir adalah pengambilan data oleh komputer berapa ketinggian air yang terukur oleh sensor ultrasonik.



Gambar 4.2 Gambar Pengukuran Sensor Ultrasonik terhadap Ketinggian Air

Keterangan Gambar 4.2.

- A : Sensor ultrasonik
- B : Penggaris
- C : Permukaan air

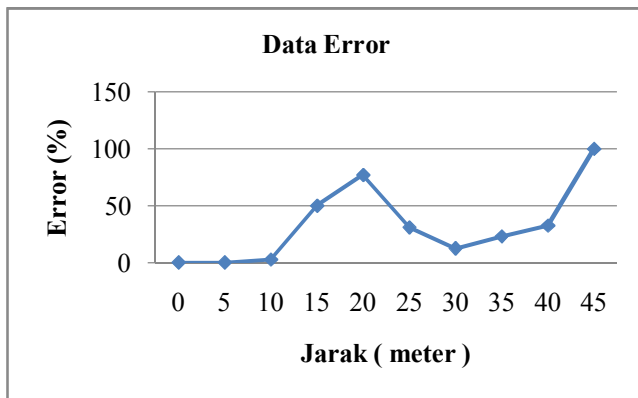
4.2. Pengujian Jarak Komunikasi Antar *Node* dan *Base*

Station

Pengujian ini dilakukan untuk mengetahui jarak yang bisa dilakukan antara *node wireless sensor* dan *base station* untuk dapat saling bertukar data. Pengujian dilakukan dengan cara memberikan jarak yang berbeda beda dengan perbedaan jarak antara pengujian sebesar 5 meter. Data akan dihitung banyaknya data yang di kirim dan data yang dibalas dari *base station* dan oleh *node*.

Tabel 4.2 Pengujian Jarak dan Error

NO	Jarak (meter)	Data Terkirim	Data Diterima	Error (%)
1	0	30	30	0
2	5	39	39	0
3	10	39	38	2.56
4	15	30	15	50
5	20	35	8	77.14
6	25	42	29	30.95
7	30	40	35	12.5
8	35	39	30	23.08
9	40	40	27	32.5
10	45	38	0	100



Gambar 4.3 Grafik *Error* Pada Komunikasi

Pengujian di lakukan dengan cara menambah jarak sebesar 5 meter dan posisi *base station* dan *node wireless sensor* berada di ketinggian 74 cm dari atas permukaan tanah. Dalam pengujian ini terlihat bahwa jarak antara *node wireless sensor* dengan *base station* sangat mempengaruhi keberhasilan untuk berkomunikasi. Jarak maksimal yang bisa di jangkau

oleh *node wireless sensor* dan base station adalah pada jarak 45 meter. Pada jarak ini komunikasi data antara *node wireless sensor* dan *base station* sudah tidak bisa berkomunikasi sama sekali. Sehingga jarak yang mungkin masih bisa berkomunikasi adalah dibawah 45 meter

4.3. Pengujian Suplai Daya

Pengujian suplai daya ini digunakan untuk mengetahui keluaran tegangan dari suplai daya. Tegangan dari suplai daya diukur pada keluaran dari suplai daya. Pengukuran menggunakan volt meter pada posisi positif menuju ke positif dari keluaran suplai daya dan negative pada posisi *ground* dari suplai daya. Gambar 4.4 menunjukkan tegangan yang terukur pada suplai daya.



Gambar 4.4 Pengukuran Tegangan Suplai Daya

Pengujian ini diharapkan suplai daya mempunyai tegangan 12 volt namun yang terukur pada volt meter bernilai 12.36 Volt. sehingga mempunyai perbedaan dengan nilai 0.36 Volt atau mempunyai perbedaan sebesar 3 %.

4.4. Pengujian Motor DC

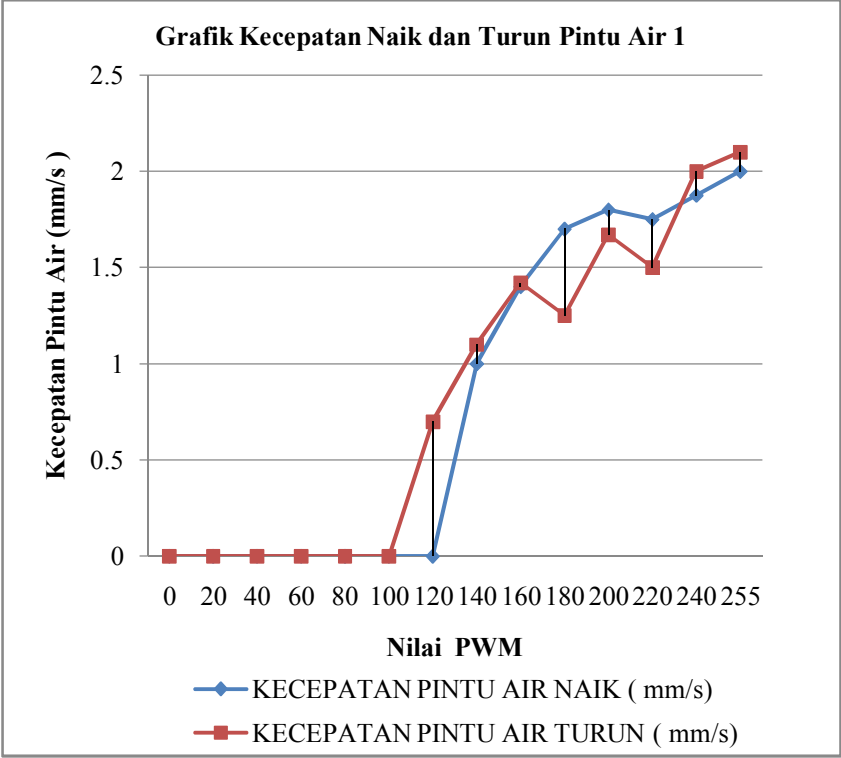
Pengujian motor DC diuji untuk mengetahui karakteristik motor DC yang digunakan sebagai pembuka dan penutup pintu air . Karakterisasi motor DC ini dimulai dengan cara memberikan keluaran tegangan berupa tegangan PWM dari arduino. Tegangan ini akan diberikan ke *driver* motor L298. Keluaran dari *driver* motor L298 tersebut akan dimasukkan ke motor DC. Sehingga PWM dari arduino akan dapat menggerakkan motor DC sesuai dengan tegangan keluaran dari PWM arduino.

Tabel 4.3 Karakteristik Motor DC pada Pintu 1

NO	NILAI PWM	NILAI TEGANGAN PWM ARDUINO (Volt)	KECEPATAN PINTU AIR NAIK (mm/s)	KECEPATAN PINTU AIR TURUN (mm/s)
1	0	0	0	0
2	20	0.4	0	0
3	40	0.79	0	0
4	60	1.18	0.6	0
5	80	1.57	0.7	0
6	100	1.95	1.3	0
7	120	2.35	1.6	0.7
8	140	2.75	1.3	1.1
9	160	3.13	2	1.42
10	180	3.5	2.2	1.25
11	200	3.92	2.14	1.67
12	220	4.25	2	1.5
13	240	4.6	2	2
14	255	4.98	2.5	2.1

Tabel 4.3 dapat dilihat bahwa semakin besar nilai tegangan PWM maka akan semakin cepat putaran motor DC yang berakibat pada kecepatan naik dan turunya pintu air. Pintu air mulai bergerak pada nilai

PWM 100 dan 120 dengan kecepatan naik 1.2 mm/s dan turun 0.7 mm/s. kecepatan tertinggi terjadi ketika nilai PWM pada arduino bernilai maksimum yaitu 255 dengan tegangan keluaranya 4.98 V. Kecepatan yang dikeluarkan oleh motor DC adalah 2.5 mm/s ketika naik dan 2.1 mm/s ketika turun.



Gambar 4.5 Grafik Kecepatan Naik dan Turun Pintu Air 1

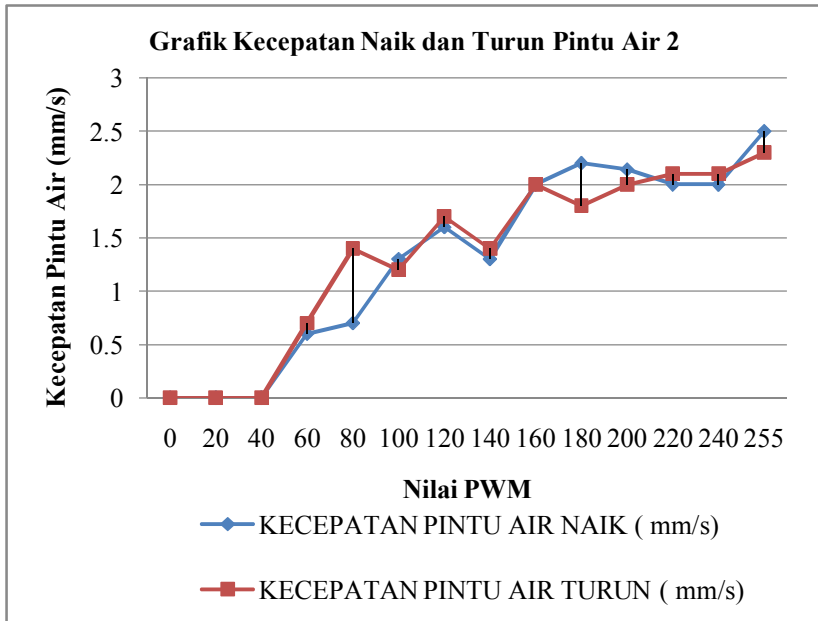
Pengujian karakteristik motor DC juga dilakukan pada pintu air 2. Pengujian dilakukan dengan cara yang sama dengan pengujian pintu air 1 yaitu dengan mengubah ubah nilai PWM agar tegangan PWM dapat berubah. Selanjutnya mengukur kecepatan pintu air naik

dan turun. Berikut hasil pengukuran nilai PWM terhadap kecepatan pintu air.

Tabel 4.4 Karakteristik Motor DC pada Pintu 2

NO	NILAI PWM	NILAI TEGANGAN PWM ARDUINO (Volt)	KECEPATAN PINTU AIR NAIK (mm/s)	KECEPATAN PINTU AIR TURUN (mm/s)
1	0	0	0	0
2	20	0.4	0	0
3	40	0.79	0	0
4	60	1.18	0.6	0
5	80	1.57	0.7	0
6	100	1.95	1.3	0
7	120	2.35	1.6	0.7
8	140	2.75	1.3	1.1
9	160	3.13	2	1.42
10	180	3.5	2.2	1.25
11	200	3.92	2.14	1.67
12	220	4.25	2	1.5
13	240	4.6	2	2
14	255	4.98	2.5	2.1

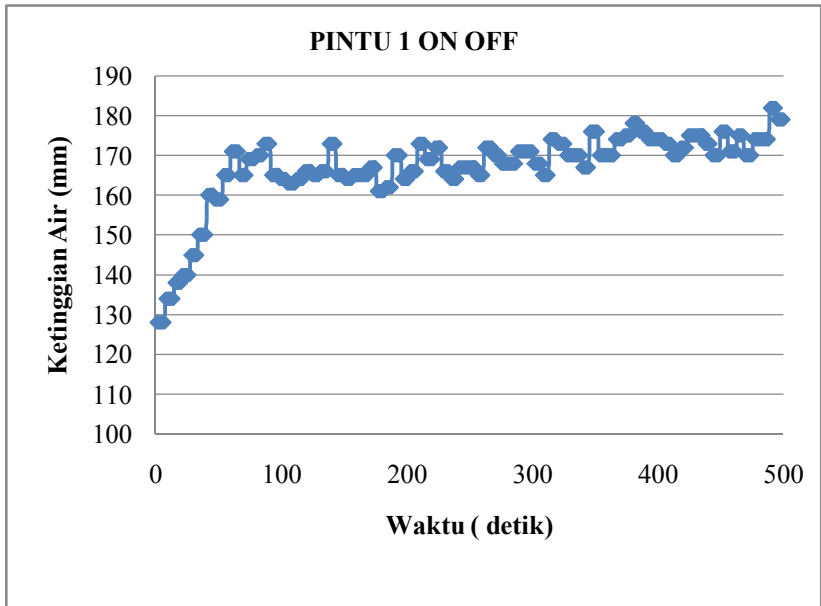
Dari tabel 4.4 dapat terlihat bahwa pintu air mulai dapat bergerak naik ketika nilai PWM pada nilai 60 dan dapat turun pada nilai PWM 120. Kecepatan pertama kali dapat bergerak adalah dengan kecepatan 0.6 mm/s ketika naik dan kecepatan 0.7 mm/s ketika turun. Kecepatan naik dan turunnya pintu air mempunyai kecepatan tertinggi pada PWM yang paling tinggi pula yaitu nilai PWM 255 dengan kecepatan naik sebesar 2.5 mm/s dan kecepatan turun 2.1 mm/s.



Gambar 4.6 Grafik Kecepatan Naik dan Turun Pintu Air 2

4.5. Pengujian Kontrol On/Off

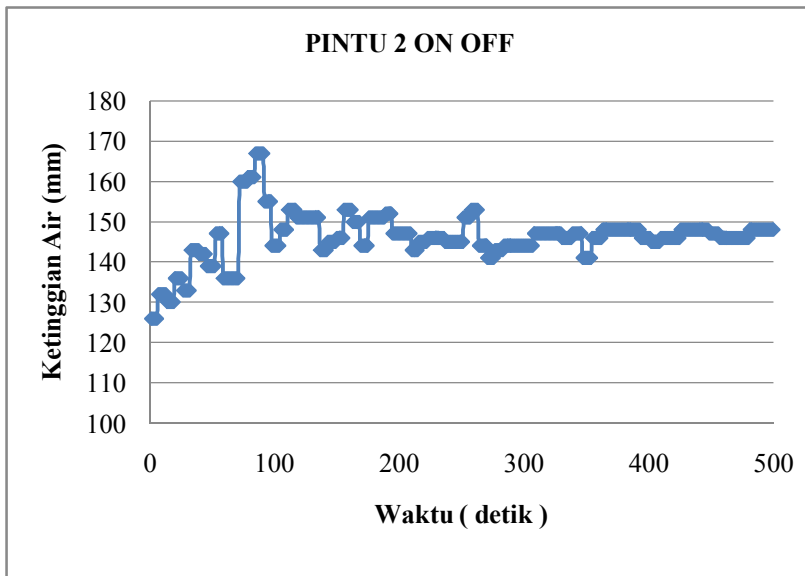
Pengujian kontrol On/Off pertama dilakukan pada pintu 1 dengan menjaga ketinggian air di ketinggian 170 mm. Pertama air di set pada ketinggian paling rendah yaitu 130 mm dan pintu air dalam keadaan tertutup. Setelah itu *node wireless sensor* dan *base station* dinyalakan serta di catat ketinggian air yang terjadi.



Gambar 4.7 Grafik Ketinggian pada Pintu Air 1

Gambar 4.7 ketinggian air mula mula pada ketinggian 130 mm lalu semakin lama semakin naik karena terisi oleh aliran air. Di ketinggian 160 mm pintu air terbuka pada posisi 1 sehingga pergerakan ketinggian air akan melambat dan pada ketinggian 170 mm pintu air lebih terbuka lagi pada posisi 2 sehingga ketinggian air langsung menurun. Namun ketinggian air belum bisa tepat pada ketinggian 170 mm. Kesalahan rata-rata pada pengujian ini sebesar 5.9 mm.

Selanjutnya pengujian pada pintu air yang ke 2 dengan cara yang sama pada pintu air 1. Pintu air 2 menjaga ketinggian air pada ketinggian 150 mm.

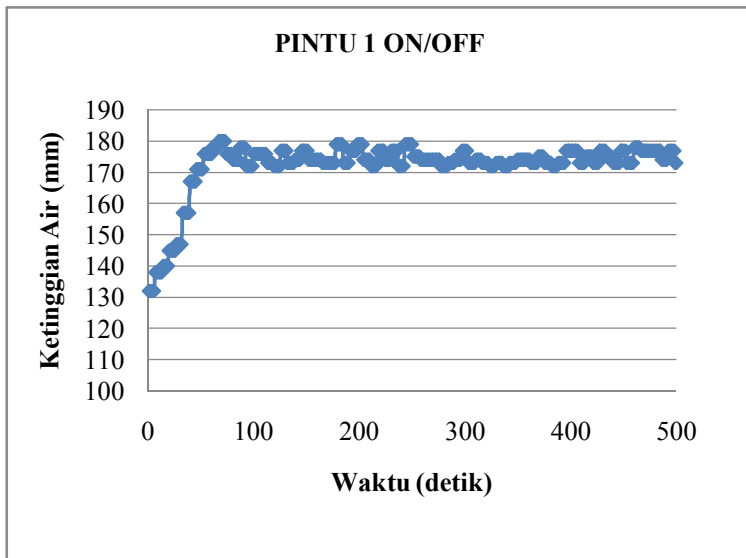


Gambar 4.8 Grafik Ketinggian pada Pintu Air 2

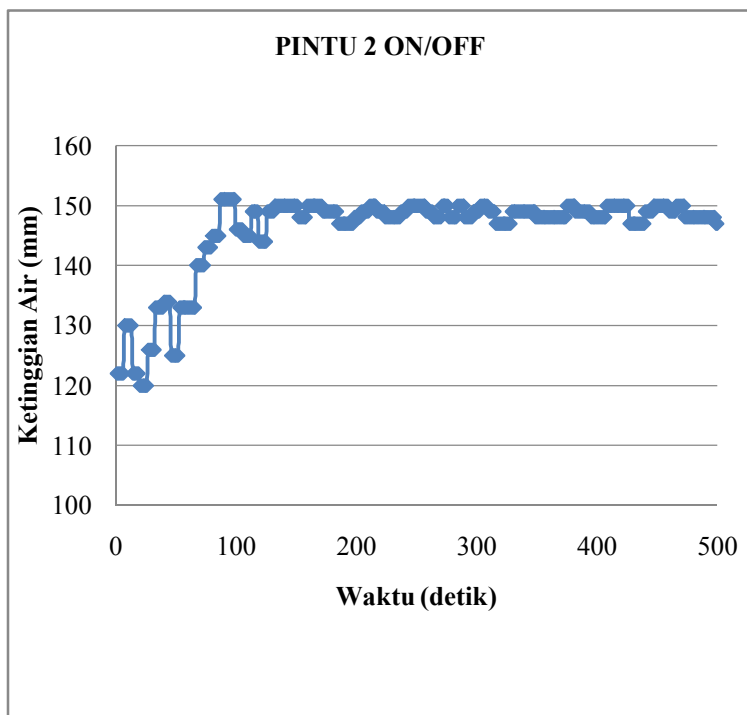
Pintu yang ke 2 ini air mula-mula di ketinggian 130 mm akan naik karena terisi oleh aliran air. Di ketinggian 150 hingga 160 pintu air terbuka pada posisi 2 menyebabkan ketinggian air menurun dan kembali lagi keposisi 1 pada ketinggian 140-150 menyebabkan ketinggian air naik secara perlahan-lahan. Kesalahan rata-rata pada pintu air 2 ini adalah 5.4 mm.

Pengujian yang terakhir merupakan pengujian yang melibatkan ke dua pintu secara bersamaan. Saat pengujian mula-mula ketinggian air diposisikan pada ketinggian 130 mm dan posisi pintu air 1 dan 2 pada kondisi tertutup lalu *node wireless sensor* dan *base station* dinyalakan. Air akan mulai mengalir memenuhi pintu air 1 dan menyebabkan ketinggian air pada pintu air 1 semakin tinggi hingga pada ketinggian tertentu pintu air terbuka dan melepas aliran air menuju ke pintu air yang kedua. Di pintu air yang ke 2 ketinggian air akan meningkat dan pada saat kondisi ketinggian tertentu maka akan terbuka pintu air untuk melepas aliran air agar mengatur ketinggian air pada titik tertentu.

Grafik yang ditunjukkan pada gambar 4.9 dan 4.10 terlihat bahwa ketinggian air dapat diatur untuk mendekati ketinggian yang diinginkan yaitu 170 mm pada pintu air 1 dan 150 mm pada pintu air 2. Pada saat ketinggian air pada ketinggian 170-180 mm pintu air terbuka pada posisi 2 sehingga ketinggian air tidak bergerak naik atau turun. Hal ini dikarenakan jumlah air yang keluar pada pintu air 1 sama dengan air yang masuk. Hal ini juga terjadi pada pintu air 2 pada ketinggian 150 mm. pintu air terbuka pada posisi 1 dan air yang masuk pada pintu air 2 sama dengan yang dikeluarkan. Dari data yang diperoleh kesalahan yang terjadi pada pengujian ini di pintu air 1 sebesar 6.4 mm sedangkan pada pintu air 2 sebesar 4.3 mm. sehingga rata-rata kesalahannya adalah sebesar 5.35 mm



Gambar 4.9 Grafik Ketinggian pada Pintu Air 1 pada Pengujian Keseluruhan



Gambar 4.10 Grafik Ketinggian pada Pintu Air 2 pada Pengujian Keseluruhan

BAB V

PENUTUP

5.1. Kesimpulan

Kesimpulan dari tugas akhir ini adalah telah dibuat *wireless sensor* terdiri atas *base station* dan *node wireless sensor* dapat berkomunikasi sejauh 40 meter. Radio frekuensi yang digunakan 433 Mhz dengan kecepatan 2000bps. Suplai daya menggunakan suplai daya yang bertegangan 12.36 Volt. Sensor ultrasonik HC-SR04 yang digunakan dapat mengukur ketinggian air secara linier pada jarak 0-200 mm. Nilai *pulse width modulation* pada arduino yang menggunakan mikrokontroler ATmega328 dengan kecepatan 16 Mhz dapat mengatur kecepatan dan arah putaran motor DC melalui driver L298. Data yang didapat dari hasil komunikasi antara *base station* dan *wireless sensor* ditampilkan menggunakan komputer berbasis Delphi 7. Kecepatan naik turun pintu air ketika nilai *pulse width modulation* maksimal sebesar 255 atau keluaran tegangan *pulse width modulation* sebesar 5V mempunyai kecepatan naik dan turun rata-rata 2 mm/s. Pintu air yang digunakan menggunakan 2 pintu air yaitu pintu air yang berada di hulu dinamakan pintu air 1 yang diatur ketinggian airnya sebesar 170 mm dan pintu air yang berada di hilir dinamakan pintu air 2 yang diatur ketinggian airnya sebesar 150 mm. Pengaturan pintu air dengan menggunakan metode on/off menghasilkan rata-rata kesalahan yang terjadi pada pintu air 1 adalah 6.4 mm dan untuk pintu air 2 yaitu 4.3 mm sehingga rata-rata kesalahan keseluruhan 5.35 mm.

5.2. Saran

Saran untuk pengembangan kedepan yang dapat dilakukan adalah menggunakan pengirim dan penerima dengan jangkauan lebih baik. Menggunakan metode multi-Hop. Sensor yang digunakan bisa lebih baik dan lebih banyak. Tampilan pada komputer dapat diperbanyak dengan software dapat merubah set point dan dapat diatur secara manual

DAFTAR PUSTAKA

1. J. Carr, Joseph. "*Sensor and Circuit*", Prentice Hall, United States of America, 1993
2. Dargie , Waltenegeus. "*Fundamentals of Wireless Sensor Network*", Wiley, United Kingdom, 2010
3. Datasheet HC-SR04, iTeaStudio.2010
4. Kadir, Abdul. "*Panduan Praktis Mempelajari Aplikasi Mikrokontroler dan Pemrogramannya Menggunakan Arduino*". Yogyakarta: Penerbit Andi Offset, 2013.
5. _____, **Analog** Write <URL: <http://www.arduino.cc/en/Reference/analogWrite>> 4 Mei 2015
6. Jihon Xia, Junqiang Lin,"*Method of Describing and Diagnosing River Health*", College of Water Conservancy & Hydropower Hohai University,2011
7. Chirag Warty, Richard Way Yu, "*Resource Allocation using ASK,FSK and PSK modulation techniques with varying M*", IEEE Aerospace Conference, 2011.

LAMPIRAN

PROGRAM BASE STATION

// Inisialisasi library, konstanta, variabel, pin input dan output

```
#include <VirtualWire.h>
```

```
const int led_pin = 13;
const int transmit_pin = 12;
const int receive_pin = 11;
const int transmit_en_pin = 3;
```

```
int
inputAA, setpointAA, errorAA, outputAA, kp, ki, kd, z, integralAA, differens
ialAA, inputAAlalu, penandapintuAA;
```

```
int
inputBB, setpointBB, errorBB, outputBB, integralBB, differensialBB, input
BBlalu, penandapintuBB;
```

```
int waktu;
word alamat, ratus, puluh, sat, sementara;
```

```
int
n, m, penanda_repeatA, penanda_repeatB, penanda_repeatAP, penanda_rep
eatBP, o, p;
```

```
void setup()
```

```
{
    Serial.begin(9600);    // Debugging only
    // Serial.println("setup");
```

```
    // Initialise the IO and ISR
```

```
    vw_set_tx_pin(transmit_pin);
```

```
    vw_set_rx_pin(receive_pin);
```

```
    vw_set_ptt_pin(transmit_en_pin);
```

```
    vw_set_ptt_inverted(true);    // Required for DR3100
```

```
    vw_setup(2000);    // Bits per sec
```

```
    vw_rx_start();    // Start the receiver PLL running\
```

```
    setpointAA = 160;    // SEET POINT PID AA
```

```

    setpointBB = 140;          // SET POINT PID BB
    pinMode(led_pin, OUTPUT);
    penandapintuBB = 1;
    penandapintuAA = 1;
}
word ketinggian_airAA, ketinggian_airBB;

void loop()
{

    word ratusmet, puluhmet, satmet;
    byte count = 1;

// Mengirim perintah ambil data dari node A

    char msgAA[7] = {'U','A','S',' ',' ',' ',' '}; //UAS
    delay(100);
    digitalWrite(led_pin, HIGH); /
    vw_send((uint8_t *)msgAA, 7);
    vw_wait_tx(); digitalWrite(led_pin, LOW);

    uint8_t bufAA[VW_MAX_MESSAGE_LEN];
    uint8_t buflenAA = VW_MAX_MESSAGE_LEN;

// Menerima data dari node A

    vw_wait_rx_max(1000);
    if (vw_get_message(bufAA, &buflenAA)) // Non-blocking
    {
        alamat = bufAA[0] + bufAA[1] + bufAA[2]

        if(alamat==235) // Alamat 235 = W + S + A== 87 +83 +65 //
        MENERIMA DATA DARI NODE A
        {
            //Serial.print(" Masuk WAS");
            n = 7;
            ketinggian_airAA = bufAA[3]*100 + bufAA[4]*10 + bufAA[5];
            penanda_repeatA = 0;
        }
    }
}

```

```

    }
    // Mengirim perintah ambil data dari node B

    char msgBB[7] = {'U','B','S',' ',' ',' ',' '};
;
    digitalWrite(led_pin, HIGH); // Flash a light to show transmitting
    vw_send((uint8_t *)msgBB, 7);
    vw_wait_tx(); // Wait until the whole message is gone
    digitalWrite(led_pin, LOW);
    delay(100);

    // Menerima data dari node B

    uint8_t bufBB[VW_MAX_MESSAGE_LEN];
    uint8_t buflenBB = VW_MAX_MESSAGE_LEN;
    vw_wait_rx_max(1000);

    if (vw_get_message(bufBB, &buflenBB)) // Non-blocking
    {
        alamat = bufBB[0] + bufBB[1] + bufBB[2];

        if(alamat==236)
        {

            m = 5;
            ketinggian_airBB= bufBB[3]*100 + bufBB[4]*10 + bufBB[5];
            penanda_repeatB = 0;
        }
    }
    // Klasifikasi Ketinggian Air

    outputAA=128;
    outputBB=128;

    if (ketinggian_airAA <= 160)
    {
        posisi_airA=0;
    }

```

```

}

if (ketinggian_airAA > 161)
{
    if (ketinggian_airAA <= 170)
    {
        posisi_airA=1;
    }
}

if (ketinggian_airAA > 171)
{
    if (ketinggian_airAA <= 180)
    {
        posisi_airA=2;
    }
}

if (ketinggian_airAA > 181)
{
    if (ketinggian_airAA <= 190)
    {
        posisi_airA=3;
    }
}

if (ketinggian_airAA > 191)
{
    if (ketinggian_airAA <= 200)
    {
        posisi_airA=4;
    }
}

if (ketinggian_airAA > 201)
{
    if (ketinggian_airAA <= 210)
    {

```

```

posisi_airA=5;
}
}
// Membandingkan klasifikasi ketinggian air dengan posisi pintu air

if (posisi_airA == posisi_pintuA)
{
outputAA = 128;
}

if (posisi_airA > posisi_pintuA)
{
outputAA = 255;
posisi_pintuA = posisi_pintuA + 1;
}
if (posisi_airA < posisi_pintuA)
{
outputAA = 0;
posisi_pintuA = posisi_pintuA - 1;
}

// Klasifikasi Ketinggian Air
if (ketinggian_airBB <= 140)
{
posisi_airB=0;
}

if (ketinggian_airBB > 141)
{
if (ketinggian_airBB <= 150)
{
posisi_airB=1;
}
}

if (ketinggian_airBB > 151)
{
if (ketinggian_airBB <= 160)

```

```

{
    posisi_airB=2;

}
}

    if (ketinggian_airBB > 161)
    {
        if (ketinggian_airBB <= 170)
        {
            posisi_airB=3;
        }
    }

    if (ketinggian_airBB > 171)
    {
        if (ketinggian_airBB <= 180)
        {
            posisi_airB=4;
        }
    }

    if (ketinggian_airBB > 181)
    {
        if (ketinggian_airBB <= 190)
        {
            posisi_airB=5;
        }
    }

    // Membandingkan klasifikasi ketinggian air dengan posisi pintu air

    if (posisi_airB == posisi_pintuB)
    {
        outputBB = 128;
    }

    if (posisi_airB > posisi_pintuB)
    {

```



```

outputBB = 255;
posisi_pintuB = posisi_pintuB + 1;

}
if (posisi_airB < posisi_pintuB)
{
outputBB = 0;
posisi_pintuB = posisi_pintuB - 1;
}

```

// Mengirim perintah ke Node

```

    ratus = outputAA/100;
    puluh = ((outputAA - (100 * ratus)) / 10);
    sat = ((outputAA - (100 * ratus)) - (10 * puluh));
    char msgAAPID[7] = {'F','A','S',ratus,puluh,sat,''};
digitalWrite(led_pin, HIGH); // Flash a light to show transmitting
vw_send((uint8_t *)msgAAPID, 7);
vw_wait_tx(); // Wait until the whole message is gone
digitalWrite(led_pin, LOW);
delay(100);

ratus = outputBB/100;
    puluh = ((outputBB - (100 * ratus)) / 10);
    sat = ((outputBB - (100 * ratus)) - (10 * puluh));

    char msgBBPID[7] = {'F','B','S',ratus,puluh,sat,'#'};

digitalWrite(led_pin, HIGH); // Flash a light to show transmitting
vw_send((uint8_t *)msgBBPID, 7);
vw_wait_tx(); // Wait until the whole message is gone
digitalWrite(led_pin, LOW);
}

delay(2000);

outputAA = 128;
ratus = outputAA/100;

```

```

puluh = ((outputAA - (100 * ratus)) / 10);
sat = ((outputAA - (100 * ratus)) - (10 * puluh));
char msgAASTOP[7] = {'F','A','S',ratus,puluh,sat,' '};

```

```

digitalWrite(led_pin, HIGH); // Flash a light to show transmitting
vw_send((uint8_t *)msgAASTOP, 7);
vw_wait_tx(); // Wait until the whole message is gone
digitalWrite(led_pin, LOW);

```

```

outputBB = 128;
ratus = outputBB/100;
puluh = ((outputBB - (100 * ratus)) / 10);
sat = ((outputBB - (100 * ratus)) - (10 * puluh));

```

```

char msgBBSTOP[7] = {'F','B','S',ratus,puluh,sat,'#'};

```

```

digitalWrite(led_pin, HIGH); // Flash a light to show transmitting
vw_send((uint8_t *)msgBBSTOP, 7);
vw_wait_tx(); // Wait until the whole message is gone
digitalWrite(led_pin, LOW);

```

```

if (ketinggian_airAA < 100)
{
    if (ketinggian_airAA >= 10)
    {
        Serial.print(0);
        Serial.print(ketinggian_airAA);
    }
}
if (ketinggian_airAA < 10)
{
    Serial.print(0);
    Serial.print(0);
    Serial.print(ketinggian_airAA);
}
if (ketinggian_airAA > 100)
{
    Serial.print(ketinggian_airAA);
}

```

```

digitalWrite(led_pin, LOW);
delay(10);

// Mengirim data ke komputer

if (ketinggian_airBB < 100)
{
    if (ketinggian_airBB >= 10)
    {
        Serial.print(0);
        Serial.print(ketinggian_airBB);
    }
}
if (ketinggian_airBB < 10)
{
    Serial.print(0);
    Serial.print(0);
    Serial.print(ketinggian_airBB);
}
if (ketinggian_airBB > 100)
{
    Serial.print(ketinggian_airBB);
}
digitalWrite(led_pin, LOW);
delay(10);

```

PROGRAM NODE A WIRELESS SENSOR

```

#define trigPin 7 // SRF Trigger pin 7
#define echoPin 8 // SRF Echo Pin 8
// inisialisasi TX RX
#include <VirtualWire.h>

const int led_pin = 13;
const int transmit_pin = 12;
const int receive_pin = 11;
const int transmit_en_pin = 3;

```

```

int nilai_perintah,nilai_pwm,penanda_repeat,n,o,asal_data,nilai_analog;
word
ratusmet,puluhmet,satmet,sementara,ketinggian_air,Aktuator_A,Aktuator_AS;
long duration,distance;
#define led 11
#define led2 10
uint8_t buf[VW_MAX_MESSAGE_LEN];
uint8_t buflen = VW_MAX_MESSAGE_LEN;
word alamat;
void setup() {
    // Setup SRF
    Serial.begin (9600);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    //Setup TX
    vw_set_tx_pin(transmit_pin);
    vw_set_rx_pin(receive_pin);
    vw_set_ptt_pin(transmit_en_pin);
    vw_set_ptt_inverted(true); // Required990 for DR3100
    vw_setup(2000); // Bits per sec
    pinMode(led_pin, OUTPUT);
    /*******
    // Setup RX
    vw_rx_start();
    //pinMode(led, OUTPUT);
    //pinMode(led2, OUTPUT);
}

byte count = 1;

void loop()
{
    if (vw_get_message(buf, &buflen)) // Non-blocking
    {
        // Menunggu perintah dari base station

        //vw_rx_stop();

```

```

    alamat = buf[0] + buf[1]; // jika alamat benar AA maka run
program dibawah
    int i;
    //Serial.print(alamat);
    if ( alamat == 150) // Header U85 + Header A65 + Header S83=
150
    {
        vw_rx_stop();
        digitalWrite(led_pin, HIGH); // Flash a light to show received good
message
        // Membaca ketinggian air

        digitalWrite(trigPin, LOW);
        delayMicroseconds(2);
        digitalWrite(trigPin, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigPin, LOW);
        duration = pulseIn(echoPin, HIGH);
        distance = ((duration/2)*10) / 22;
        distance = 270 -( distance ); // milimeter

        sementara = distance;
        sementara = distance+ 123;
        ratusmet = sementara/100;
        puluhmet = ((sementara - (100 * ratusmet)) / 10);
        satmet = ((sementara - (100 * ratusmet)) - (10 * puluhmet));

        // Kirim data ke base station

        Serial.print("KIRIM WSA");

        //delay(100);
        char msg[7] = {'W','S','A',ratusmet,puluhmet,satmet,' '}; //
        delay(200);
        msg[6] = count;
        digitalWrite(led_pin, HIGH); // Flash a light to show transmitting
        vw_send((uint8_t *)msg, 7);
        vw_wait_tx(); // Wait until the whole message is gone
        digitalWrite(led_pin, LOW);

```

```

    Serial.print(" Kirim data WSA =");
    Serial.print(distance);
    Serial.print("=");
    //delay(100);
    vw_rx_start();
}
// Implementasi aktuator

if ( buf[0] == 70) // Header F70 = 70 // PROSES PID
{
    //alamat = buf[1];
    Serial.print(alamat);
    if(buf[1] == 65)
    {
        vw_rx_stop();
        // PIN 4 UNTUK MAJU
        // PIN 2 UNTUK MUNDUR
        Aktuator_A = buf[3]*100 + buf[4]*10 + buf[5];

        Serial.print("buf=");
        Serial.print(buf[3]);
        Serial.print(buf[4]);
        Serial.print(buf[5]);
        Serial.print("=");

        Serial.print("AKTUATOR A =");
        Serial.print(Aktuator_A);
        Serial.print("=");
        if(Aktuator_A > 127)
        {
            digitalWrite(4,HIGH); // PPINTU AIR NAIK
            digitalWrite(2,LOW);
            Aktuator_AS = ((Aktuator_A - 127) * 2 ); //Aktuator_A mulai
dari 0-255
            Aktuator_AS = Aktuator_AS;
            if (Aktuator_AS > 255)
            {
                Aktuator_AS = 255;
            }

```

```

if (Aktuator_AS < 0)
{
Aktuator_AS = 0;
}

analogWrite(6,Aktuator_AS);
Serial.print("NAIK=");
Serial.print(Aktuator_AS);
Serial.print("=");
}

if(Aktuator_A < 126)
{
digitalWrite(2,HIGH); // PINTU AIR TURUN
digitalWrite(4,LOW);
Aktuator_AS = ((126 - Aktuator_A) * 2); //Aktuator_A mulai dari
0-255
Aktuator_AS = Aktuator_AS;
if (Aktuator_AS > 255)
{
Aktuator_AS = 255;
}
if (Aktuator_AS < 0)
{
Aktuator_AS = 0;
}

nilai_analog = 0;
nilai_analog = analogRead(5);

if (nilai_analog > 100)
{
analogWrite(6,0); // pintu air berhenti
Serial.print("pintu air berhenti=");
Serial.print(Aktuator_AS);
Serial.print("=");
}

```

```

        if (nilai_analog < 100)
        {
            analogWrite(6,Aktuator_AS);
            Serial.print("TURUN=");
            Serial.print(Aktuator_AS);
            Serial.print("=");
        }
    }
    vw_rx_start();
}
}
digitalWrite(led_pin, LOW);

}
}

```

PROGRAM NODE B WIRELESS SENSOR

```

#define trigPin 7 // SRF Trigger pin 7
#define echoPin 8 // SRF Echo Pin 8
// inisialisasi TX RX
#include <VirtualWire.h>

const int led_pin = 13;
const int transmit_pin = 12;
const int receive_pin = 11;
const int transmit_en_pin = 3;
int nilai_perintah,nilai_pwm,asal_data;
word
ratusmet,puluhmet,satmet,sementara,ketinggian_air,Aktuator_B,Aktuator_BS;
long duration,distance;
uint8_t buf[VW_MAX_MESSAGE_LEN];
uint8_t buflen = VW_MAX_MESSAGE_LEN;
word alamat;
void setup() {
    // Setup SRF
    Serial.begin (9600);
    pinMode(trigPin, OUTPUT);

```



```

pinMode(echoPin, INPUT);
//Setup TX
vw_set_tx_pin(transmit_pin);
vw_set_rx_pin(receive_pin);
vw_set_ptt_pin(transmit_en_pin);
vw_set_ptt_inverted(true); // Required990 for DR3100
vw_setup(2000); // Bits per sec
pinMode(led_pin, OUTPUT);
//*****
// Setup RX
vw_rx_start();
//pinMode(led, OUTPUT);
//pinMode(led2, OUTPUT);
}

byte count = 1;

void loop() {
  // Receive Data RX
  // Menunggu perintah dari base station

  vw_wait_rx();
  if (vw_get_message(buf, &buflen)) // Non-blocking
  {
    alamat = buf[0] + buf[1]; // jika alamat bener AA maka run
    program dibawah
    int i;

    if ( alamat == 151) // Header U85 + Header B66 = 151
    {
      vw_rx_stop();
      digitalWrite(led_pin, HIGH

      //word alamat;
      // Membaca ketinggian air

      digitalWrite(trigPin, LOW);
      delayMicroseconds(2);
      digitalWrite(trigPin, HIGH);

```

```

delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = ((duration/2)*10 / 22) ;
distance = 270 - (distance);

```

// Kirim data ke base station

```

sementara = distance;
ratusmet = sementara/100;
puluhmet = ((sementara - (100 * ratusmet)) / 10);
satmet = ((sementara - (100 * ratusmet)) - (10 * puluhmet));
    delay(100);
    Serial.print("KIRIM WSB");

    //delay(100);
    char msg[7] = {'W','S','B',ratusmet,puluhmet,satmet,''}; //
Mengirim data
    delay(10);
    msg[6] = count;
    digitalWrite(led_pin, HIGH); // Flash a light to show transmitting
    vw_send((uint8_t *)msg, 7);
    vw_wait_tx(); // Wait until the whole message is gone
    digitalWrite(led_pin, LOW);
    Serial.print(" Kirim data WSB =");
    Serial.print(distance);
    Serial.print("=");
    //delay(150);
    vw_rx_start();
}

```

// Implementasi aktuator

```

if ( buf[0] == 70) // Header F70 = 70    {
    //alamat = buf[1];
    Serial.print(alamat);
    if(buf[1] == 66)    {
        // PIN 4 UNTUK MAJU
        // PIN 2 UNTUK MUNDUR
        vw_rx_stop();
    }
}

```

```

    Aktuator_B = buf[3]*100 + buf[4]*10 + buf[5];

    Serial.print("buf=");
    Serial.print(buf[3]);
    Serial.print(buf[4]);
    Serial.print(buf[5]);
    Serial.print("=");

    Serial.print(" AKTUATOR B =");
    Serial.print(Aktuator_B);
    Serial.print("=");
    if(Aktuator_B > 127)
    {
        digitalWrite(4,HIGH); // PPINTU AIR NAIK
        digitalWrite(2,LOW);
        delay(10);
        Aktuator_BS = ((Aktuator_B - 127) * 2 ); //Aktuator_A mulai dari
0-255
        Aktuator_BS = Aktuator_BS + 30;

        if (Aktuator_BS > 255)
        {
            Aktuator_BS = 255;
        }
        if (Aktuator_BS < 0)
        {
            Aktuator_BS = 0;
        }

        analogWrite(6,Aktuator_BS);
        Serial.print("NAIK=");
        Serial.print(Aktuator_BS);
        Serial.print("=");

    }
    if(Aktuator_B < 126)
    {
        digitalWrite(2,HIGH); // PINTU AIR TURUN

```

```

    digitalWrite(4,LOW);
    delay(10);
    Aktuator_BS = ((126 - Aktuator_B) * 2 ); //Aktuator_A mulai dari
0-255
    Aktuator_BS = Aktuator_BS + 30;
    if (Aktuator_BS > 255)
    {
        Aktuator_BS = 255;
    }
    if (Aktuator_BS < 0)
    {
        Aktuator_BS = 0;
    }

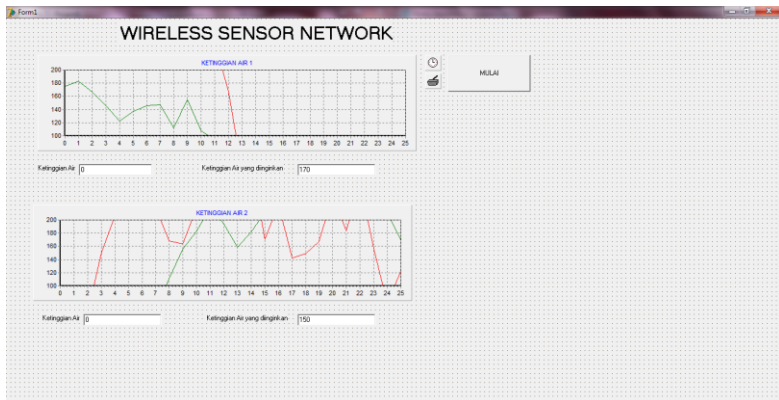
    analogWrite(6,Aktuator_BS);
    Serial.print("TURUN=");
    Serial.print(Aktuator_BS);
    Serial.print("=");

    }
    vw_rx_start();
    }
    }
    digitalWrite(led_pin, LOW);
}
}

```

PROGRAM DELPHI

- *User Interface*



- **Perangkat lunak Delphi**

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,

Dialogs, CPort, CPortCtl, TeEngine, Series, StdCtrls, ExtCtrls,

TeeProcs,

Chart;

type

TForm1 = class(TForm)

ComPort1: TComPort;

Button1: TButton;

Chart1: TChart;

Chart2: TChart;

Label1: TLabel;

Series1: TLineSeries;

Series2: TLineSeries;

Edit1: TEdit;

Timer1: TTimer;

Edit2: TEdit;

Label4: TLabel;

```

Edit3: TEdit;
Label2: TLabel;
Label5: TLabel;
Label6: TLabel;
Edit5: TEdit;
Series3: TLineSeries;
Series4: TLineSeries;
procedure Button1Click(Sender: TObject);
procedure ComPort1RxChar(Sender: TObject; Count: Integer);
procedure delay(lama:longint);
procedure Timer1Timer(Sender: TObject);
//procedure TForm1.ComPort1RxChar(Sender: TObject; Count:
Integer);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form1: TForm1;
    x,k,n          : integer;
    grafikx,dummy,nilaiAA,nilaiBB,nilaierror1,nilaierror2    : extended;
    Str            : string;
    buffer : array[1..12] of real;
    //grafikx      : Extended;
implementation

{$R *.dfm}

procedure TForm1.delay(lama:longint);
var ref:longint;
begin
    ref:=gettickcount;
    repeat application.ProcessMessages;
    until ((gettickcount-ref)>=lama);
end;

procedure TForm1.Button1Click(Sender: TObject);

```

```

begin
ComPort1.ShowSetupDialog;
ComPort1.Open;
n:=1;
Edit3.Text:=inttostr(160); // Set Point 1
Edit5.Text:=inttostr(140); // Set Point 2
end;

procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
begin

ComPort1.ReadStr(Str,Count);
//delay(2000);
//Edit1.Text:=Str;
dummy:=StrToFloat(Str);
buffer[n]:=dummy;
n:=n+1;
//dummy:=Edit1.Text;
//delay(500);
//grafikx:=StrToFloat(dummy);
//grafikx := 0;

//ComPort1.WriteStr(Str); // Buat Ngrim data serial

end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin

n:=1;
nilaiAA := buffer[1]*100 + buffer[2]*10 + buffer[3];
nilaiBB := buffer[4]*100 + buffer[5]*10 + buffer[6];
//nilaierror1 := buffer[7]*100 + buffer[8]*10 + buffer[9];
//nilaierror2 := buffer[10]*100 + buffer[11]*10 + buffer[12];
Edit1.Text:=FloattoStr(nilaiAA);
Edit2.Text:=FloattoStr(nilaiBB);

```

```
Series1.AddXY(k,nilaiAA);  
Series3.AddXY(k,160);  
Series2.AddXY(k,nilaiBB);  
Series4.AddXY(k,140);  
k:=k+1;  
end;  
  
end.
```


RIWAYAT HIDUP PENULIS



Burhanudin Rabani lahir di Surabaya 13 September 1992. Merupakan putra ke 3 dari 3 bersaudara pasangan bapak Muljo dan Ibu Sumiasih. Penulis menempuh jenjang pendidikan dari SDN Bungurasih 1, SMP Negeri 1 Taman dan SMA Negeri 1 Sidoarjo. Pada tahun 2011 penulis melanjutkan studinya di Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember. Pada tahun ke tiga penulis mengambil bidang studi elektronika.

Email : burhanudinrabani92@gmail.com